AFRL-RY-WP-TR-2013-0172

MSEE: STOCHASTIC COGNITIVE LINGUISTIC BEHAVIOR MODELS FOR SEMANTIC SENSING

Hong Man and Yu-dong Yao

Stevens Institute of Technology

SEPTEMBER 2013
Final Report

Approved for public release; distribution unlimited.
See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RY-WP-TR-2013-0172 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//Signature//	//Signature//
KELLY MILLER, Program Manager Assessment and Integration Branch Layered Sensing Exploitation Division	CHRISTINA SCHUTTE, Branch Chief Assessment and Integration Branch Layered Sensing Exploitation Division

//Signature//

DOUG HAGER, Deputy
Layered Sensing Exploitation Division
Sensors Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)* September 2013	**2. REPORT TYPE** Final	**3. DATES COVERED *(From - To)*** 13 September 2011 – 30 July 2013

4. TITLE AND SUBTITLE

MSEE: STOCHASTIC COGNITIVE LINGUISTIC BEHAVIOR MODELS FOR SEMANTIC SENSING

5a. CONTRACT NUMBER

FA8650-11-1-7152

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

61101E

6. AUTHOR(S)

Hong Man and Yu-dong Yao

5d. PROJECT NUMBER

1000

5e. TASK NUMBER

5f. WORK UNIT NUMBER

Y02C

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Stevens Institute of Technology
1 Castle Point Terrace
Hoboken, NJ 07030

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-RY-WP-TR-2013-0172

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)

AFRL/RYAA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)

AFRL-RY-WP-TR-2013-0172

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

Report contains color.

14. ABSTRACT

This report summarizes the major findings from our research on a semantic information representation framework (SIRF) for visual sensing scenarios. First, the concept and architecture of a cognitive linguistic (CL) based SIRF is introduced. Two levels of information abstraction are proposed within this framework. At the syntactic level, a probabilistic contest free grammar (PCFG) method is employed for information compression and summarization. At the semantic level, a Bayesian network approach is used to achieve semantic concept inference and reasoning. To facilitate the functions of this SIRF, several conceptual primitive modeling methods are proposed, which include a dynamic structure preserving map (DSPM) for individual human action recognition, a Gaussian Process Dynamic Model with Social Network Analysis (GPDM-SNA) for a small human group action recognition, an extended GPDM-SNA method for human object interaction (HOI) recognition, and a pyramid histogram of gradient (pHOG) method for human object recognition based on gait images. In addition to these conceptual primitive models, two quantities sensing modality utility assessment methods are introduced. They are essentially feature selection methods, one is based sparse imputation and one is based on 1_1 graph. Extensive experiments on publicly available datasets have been conducted to assess the effectiveness of the proposed methods, and highly competitive and promising results have been observed.

15. SUBJECT TERMS

semantic information representation, cognitive linguistics, probabilistic context free grammar, Bayesian network, sparse coding, group action recognition, human object interaction, gait recognition, sensor utility metrics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON (Monitor)**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	114	Kelly Miller **19b. TELEPHONE NUMBER *(Include Area Code)*** N/A

MSEE: Stochastic Cognitive Linguistic Behavior Models for Semantic Sensing

Final Project Report

Contract Number: FA8650-11-1-7152

Principle Investigators: Hong Man and Yu-dong Yao

Department of ECE
Stevens Institute of Technology

07/30/2013

Contents

1. Introduction 4

2. Semantic Information Representation Framework 8

2.1. Cognitive Linguistics Based Semantic Information Representation Framework 8

2.1.1. Cognitive Linguistics and Semantic Sensing 8

2.1.2. Layered Architecture of SIRF 9

2.1.3. Implementation 11

2.2. Learning of Syntactic Constructs in SIRF, a Probabilistic Context Free Grammar Approach 14

2.2.1. Introduction 14

2.2.2. PCFG in Visual Sensing 15

2.2.3. Proposed framework 16

2.2.4. Merge of Things 17

2.2.5. Hierarchical Grammar Rule Induction 18

2.2.6. Video Event Parsing 21

2.2.7. Experimental Results 21

2.3. Semantic Reasoning in SIRF, a Bayesian Network Approach 23

2.3.1. Bayesian Network in SIRF 23

2.3.2. Construction of BN for SIRF Concept Models 24

2.3.3. Learning Local Probability Distributions 25

2.3.4. A BN SMC Use Case 27

3. Primitive Modeling in SIRF 31

3.1. Dynamic Structure Preserving Map (DSPM) Method for Individual Human Action Primitive Modeling 31

3.1.1. Introduction 31

3.1.2. Related Works 32

3.1.3. Dynamic Structure Preserving Map for Action Recognition 33

3.1.4. Adaptive Merging Strategy for Clustering Optimization 38

3.1.5. Dynamic Model for Action Recognition 43

3.1.6. Experiments and Performance 44

3.2. Small Human Group Activity Modeling Based on Gaussian Process Dynamic Model and Social Network Analysis (SN-GPDM) 48

3.2.1. Introduction 48

3.2.2. Social Network Analysis Based Feature Set 51

3.2.3. Gaussian Process Dynamical Model and Conditional GPDM 53
3.2.4. Proposed Framework for Behavior Classification 56
3.2.5. Experimental Results 58
3.3. Human object interactions using SN-GPDM 61
3.3.1. Introduction 61
3.3.2. Human and Object Detection and Tracking 63
3.3.3. Social Network Analysis based HOI feature 64
3.3.4. Weighted Social network 66
3.3.5. Experimental Results 66
3.4. Human Gait Recognition Based on Pyramid Histogram of Oriented Gradients (pHOG). 69
3.4.1. Introduction 69
3.4.2. Framework Overview 70
3.4.3. Experimental Results 74
4. Quantitative Metrics of Semantic Utility 76
4.1. Feature Selection via Sparse Imputation (FSSI) 76
4.1.1 Introduction 76
4.1.2. Related Work 77
4.1.3. FSSI METHOD 80
4.1.4. Experimental Results 82
4.2. ℓ_1Graph Based on Sparse Coding for Feature Selection 89
4.2.1. Introduction 89
4.2.2. Feature Score Based on ℓ_1Graph 89
4.2.3. Experimental evaluation 91
5. Conclusions and Discussions 96
5.1. Conclusions 96
5.1.1. Semantic Information Framework 96
5.1.2. Primitive Modeling 96
5.1.3. Sensor Utility Metrics 97
5.2. Discussions 98
5.3. Summary of achievements on SOW tasks: 99
Publications: 101
References 103

1. Introduction

Bridging the semantic gap between data centric sensor processing outcomes and objective driven situation awareness requirements remains a major challenge in developing an effective sensing, exploitation and execution (SEE) system. To address this issue, we introduced a semantic information representation framework (SIRF) capable of describing situational and behavioral semantics in a complex visual environment. This framework consists of a layered architecture of information representations inspired by cognitive linguistics principles.

Cognitive linguistics [1][2][3][4][5] is an emerging theory of human language acquisition, and it claims that the nature of semantics as the intersection between different languages can be described using five conceptual primitives, i.e. "things", "places", "paths", "actions" and "causes" [1][3][6]. Within visual sensing domain, cognitive linguistics provides helpful insights on semantic linkages between human languages (i.e. queries) and sensor languages (i.e. visual sensor data processing outputs). Compared with ontology-based semantic representations [7], cognitive linguistics is more natural for the expression of complex compositions of relations and logics in a common sensing scenario. It enables autonomously generated cross-modality heuristics for discovering and describing semantic concepts.

Theories of natural languages production generally agree that there are three independent levels of text representations, i.e. lexical level, syntactic level, and semantic level [8]. The proposed semantic information representation framework (SIRF) closely resembles this structure. It consists of three components: 1) conceptual primitive definitions and extractions, 2) syntactic parsing, and 3) semantic reasoning.

Our semantic representation framework is constructed around the cognitive linguistic conceptual primitives. On one hand, these primitives are still at a relatively low conceptual level, so that direct and automatic inference from raw sensor inputs is possible. On the other hand, according to cognitive linguistics, these primitives are sufficient building blocks for complex semantic concepts that can be modeled through sensorimotor metaphor. Therefore our proposed representation framework provides an appropriate intermediate conceptual layer that bridges high level semantics with low level sensor data.

Considering each sensor output as a document, conceptual primitives are basic “words” from a structured vocabulary. Numerous stochastic models and machine learning methods already exit for extracting these conceptual primitives from sensor outputs, which makes SIRF feasible.

At syntactic level, a new conceptual primitive can be iteratively constructed from a series of lower level primitives. This process is modeled through Probabilistic Context Free Grammar (PCFG) [9][10] in SIRF. Context free grammar is a commonly used in modeling phrase structures; and it is more flexible than regular grammars (equivalent to HMMs) in the Chomsky hierarchy. The definitions of cognitive linguistic conceptual primitives provide SIRF a set of inherent PCFG production rules. Furthermore, additional production rules can be learned from data based on minimum description length (MDL) criteria. With these production rules, higher level primitives (i.e. phrases) can be constructed from lower level primitives (i.e. words) through “merge” and “construct” operations. Given a set of production rules associated with different concepts, string parsing tools, such as Earley parser, can be used to parse unknown data sequences, and find the most likely production rule set to predict the concept of the sequence.

After syntactic parsing, Bayesian Networks (BNs) can be constructed to perform semantic reasoning among the conceptual primitives across multiple abstraction levels. BNs have shown considerable success in describing causal semantics in expert systems [11]. Within SIRF, each BN node represents a single primitive that is relevant to a concept of interest. The structures of such BNs are usually determined by expert knowledge. A unique advantage of using cognitive linguistics primitives in SIRF is that, for human operators, the causal relationships of these BN nodes are intuitive to identify, because they resembles human cognitive experience. The conditional probabilities are normally estimated from data. Given a BN structure and a collection of multi-modal sensor inputs and social media feeds, maximum likelihood estimation or expectation maximization (EM) methods can be used to calculate the conditional probabilities of the primitives. Once a BN is constructed, inference queries can be evaluated through marginalization. A top-down reasoning can provide predictive support for primitive nodes based on concept queries; and a bottom-up reasoning can provide diagnostic support for concept nodes based on evidences from primitive nodes.

To support the proposed SIRF, we introduced a series of new primitive modeling tools and methods. They are complementary to many existing modeling methods. In particular, we have focused on several necessary but not well studied topics, including individual human action modeling, small human group action modeling, human-object interaction modeling and human object recognition over a distance.

Action primitive plays an important role in semantic representation. On human action recognition, we introduced a Dynamic Structure Preserving Map (DSPM) to model human actions directly from coarse optical flow fields, which is inspired by the latest feature learning paradigm. In this method, we modified and improved the adaptive learning procedure in self-organizing map (SOM) and then captured the dynamics of best matching neurons through Markov random walk. DSPM can learn implicit spatial-temporal correlations from optical flow feature sets and preserve the intrinsic topologies characterized by different human motions. Experimental results showed that this method can achieve highly competitive action recognition performance across a wide range of test video datasets.

Action of small human group is one level higher than individual human action. Group action is not a mere collection of individual actions in a vector space. It represents one more level of primitive abstraction, which can be incorporated in the syntactic parsing in SIRF. Compared to single human activity recognition, group human activity recognition has more challenges, such as varying group size, the varying time duration, mutual occlusions between different people, and the interaction within or between groups. We proposed a novel structural feature set to represent group behavior as well as a probabilistic framework for group activity learning and recognition. We first apply a robust multiple targets tracking algorithm to track each individual in the entire image region. Small groups are then clustered based on the output positions of the tracker. After that, we introduce a set of social network analysis (SNA) based structural features to describe the dynamic behavior of small group people in each frame. A Gaussian Process Dynamical Model (GPDM) is then employed to learn the temporal activity of small group people overtime. After training, the new group activity will be identified by computing the conditional probability with each learned GPDM. Our experimental results indicate that our proposed features and behavior model can successfully capture both the spatial and temporal dynamics of group people behavior, and correctly identify different group activities.

Similar to human group action recognition, human-object interaction (HOI) recognition is also a necessary but challenging task in computer vision. In sophistic HOI scenarios, the major difficulty is the irregular motions of human body parts and objects of interest. Again, an interaction concept can not be represented as a collection of the individual motions of body parts and objects. We extended the structure preserving SNA based feature set to describe the relationships and motion distributions of various body parts and objects. In this approach, the detected human body parts and objects are treated as nodes in social network graphs, and a set of SNA features including *closeness*, *centrality* and *centrality with relative velocity* are extracted for action recognition. To further adapt SNA for HOI, we introduced a weighted social network structure, in which the nodes representing the object of interest and some body parts are weighted more than the others in SNA feature calculation. A major advantage of the SNA based feature set is its robustness to varying node numbers and erroneous node detections, which are very common in human-object interactions. An SNA feature vector will be extracted for each frame and different human-object interactions are classified by two classification methods, including Support Vector Machine (SVM) and Hidden Markov Model (HMM). The experimental results on four different human-object interactions from HMDB dataset demonstrated that the proposed method can effectively capture the dynamical characteristics of human-object interaction and outperforms the state of art methods in human-object interaction recognition.

On *thing* primitive extraction, although we tend to use existing methods, we realize that in most surveillance scenarios, human objects are far away from camera, and a recognition using appearance features is usually unreliable. As an uncommon biometric modality, human gait recognition has a great advantage of identify people at a distance without high resolution images. We introduced a human gait recognition framework that consists of a reliable background subtraction method followed by the pyramid of Histogram of Gradient (pHOG) feature extraction on the silhouette image, and an HMM based classifier. Through background subtraction, the silhouette of human gait in each frame is extracted and normalized from the raw video sequence. After removing the shadow and noise in each region of interest (ROI), pHOG feature is computed on the silhouettes images. Then the pHOG features of each gait class will be used to train a corresponding HMM. In the test stage, pHOG feature will be extracted from each test sequence and used to calculate the posterior probability toward each trained HMM model. Experimental results on the CASIA Gait Dataset B demonstrate that with our proposed method can achieve very competitive recognition rate.

In addition, we have also explored sparse representation based sensor or sensor modality effective utility assessment methods. Given an observation vector containing all sensor outputs, sensor effective utility assessment is equivalent to feature selection in machine learning. Feature selection, which aims to obtain most informative feature subsets, has been an active research topic for many years. A critical task in designing a feature selection method is to define an effective feature evaluation metric. In this work, we selected the imputation quality in sparse representation as the evaluation metric. Sparse imputation is a technique to achieve the best sparse representation quality in classification tasks when one or more features are missing. In the proposed feature selection via sparse imputation (FSSI) method, we test each individual feature by removing it from the feature and then evaluate the sparse representation quality. The higher the representation quality indicates the lower utility from the selected feature. This feature selection method is evaluated in classification tasks. Comparative studies are conducted with existing feature selection methods (such as Fisher score and Laplacian score). Experimental results on benchmark data sets demonstrate the effectiveness of FSSI method.

Since sparse coding can be represented in ℓ_1 graphs, we further extended the FSSI idea into a general ℓ_1 graph based feature selection method. In this work, we propose a "filter" method for unsupervised feature selection which is based the geometry properties of ℓ_1 graph. ℓ_1 graph is constructed through sparse coding, and it establishes the relations of feature subspaces. The quality of features is evaluated by features' local preserving ability. We compare our method with classic unsupervised feature selection methods (Laplacian score and Pearson correlation) and supervised method (Fisher score) on benchmark data sets. The classification results based on support vector machine, k-nearest neighbors and multi-layer feed-forward networks demonstrate the efficiency and effectiveness of our method.

The remaining sections of this report are organized as follows.

In Section 2, the proposed semantic information representation framework is presented in details. It contains three subsections. The first subsection introduces the main concepts and architecture of cognitive linguistics based SIRF. The second subsection presents syntactic parsing based on PCFG. The third subsection provides Bayesian network based SIRF concept models.

In Section 3, several conceptual primitive modeling methods are presented. In the first subsection, a dynamic structure preserving map (DSPM) is introduced for individual human action recognition. In the second subsection, a small human group action recognition based on Gaussian Process Dynamic Model and Social Network Analysis is introduced. In the third subsection, the extended GPDM-SNA method is applied to human object interaction recognition. In the forth subsection, a human silhouette extraction method and a pyramid HOG feature is introduced to recognize human object based on gait images.

In Section 4, conclusions and discussions on the project findings are presented. Publications related to this project are listed at the end of the report.

2. Semantic Information Representation Framework

2.1. Cognitive Linguistics Based Semantic Information Representation Framework

2.1.1. Cognitive Linguistics and Semantic Sensing

Semantics refers to the meaning of a sign such as the meaning of a word within a language. Semantics-based techniques have been applied in many application domains, e.g., semantic web, semantic sensor networks and semantic database. Currently the most accessible semantic representation approach is through ontology. The web ontology language (OWL) [7] is well developed in the semantic web, but falls short as a semantic sensing representation language. Specifically, OWL forces one to represent all properties into ontology or rules, while natural language and human reasoning (e.g., metaphor) do not. OWL requires unified ontologies which are unnatural in knowledge domains where each community of interest develops its own dialect and domain-specific ontologies causing concept incompatibilities at the domain intersections, where common things, actions, and causal reasoning facilitate human understanding.

Cognitive linguistics is the study of human language in terms of neonatal development and evolution in human physiology, sociology, and psychology [1][2]. It offers a different perspective for knowledge representation. Cognitive linguists argue convincingly [4][5] from neonatal development that language is an index into prior sensorimotor experience, and the primary primate reasoning mechanism is metaphor, the binding of new experiences and words to existing personal sensorimotor experiences and related words.

George Lakoff laid foundations of cognitive linguistics in his classic 1987 text [2], in which he showed how categories structure language and thought, with categories organized around the most common experiences of the group that employ the language. Peter Gärdenfors [5] and others refined Lakoff's early categories into what may be called relatively orthogonal categories. One may isolate five relatively orthogonal categories (therefore "dimensions") that circumscribe the conceptual primitives of cognitive linguistics: *thing*, *place*, *path*, *action* and *cause*, which are evident in common experience of the domain and the primitive semantic components [1][3][6]. The application of these conceptual primitives in semantic sensing domain is illustrated below.

- ***Thing***: In cognitive linguistics, *things* invariably are nouns that are short, simple, and universally understood. They are the fundamental gestalts for reasoning. In semantic sensing domain, *things* would be the most common objects, such as objects of interest (e.g. human object, vehicle). *Things* are basic components to present physical entities of a scene. Sensing experts are equally familiar with the semantics of these basic components.

- ***Place***: In cognitive linguistics, *places* are opportunities for *things* to have sensorimotor interaction with referential *things*, so *place* defines a vector field about the reference object with named subspaces. *Places* in semantic sensing would be those vector fields that hold *things*, such as the sensed environment or background associated with the objects of interest, or the time duration of a certain event. A *thing* may have different meanings in different *places*, therefore *place* provides essential contextual information in sensing.

- ***Path***: A *path* is a sequence of *places*, which denotes a configuration of related *places*. The cognitive linguistics notion of path organizes *places* into the structure of an event. A *path* is a precognitive gestalt that organizes a set of *places* into a coherent whole, which would provide a series of processing and achieve an aim. In semantic sensing, a *path* may represent the trajectories of moving objects, or the time series of certain events, or a simple spatial/temporal container of a series of *places*.

- ***Action***: In cognitive linguistics, an *action* is the basic component to present behavior or process, which expresses that a *thing* takes some action in its *place* or moves along a *path*. In semantic sensing, *action* may represent the motion behavior of objects.

- ***Cause***: *Causes* are *things* that set other *things* in motion and constrain actions. *Causes* may lead to an event in which multiple things participate. In semantic sensing, *causes* are particularly important in describing event-driven scenarios. They may be absent in continuous surveillance and monitoring missions.

Semantic sensing based on cognitive linguistics is thing-centric, which starts with the abstraction of common things of the sensing domain, and then adds places, action and paths built on these things. Cognitive linguistics modeling of the semantic sensing domains reflects the richness of behaviors experienced by the objects of interest, as they interacting with other objects as well as the environment.

Many stochastic models and machine learning methods already exit for extracting these conceptual primitives from sensor outputs. Object detection and recognition techniques are quite mature in many sensing modalities. Sensed environment can be identified through certain background modeling techniques, or maybe readily available during sensor deployment. Many vision, acoustic, RF and laser techniques have been developed to generate 2D or 3D trajectories of moving objects. Action or motion behavior detection and recognition have been popular research topics in recent years. Cause of actions is a relatively complicated issue and may not be directly observable from sensor outputs. However certain heuristics can be applied to identify simple and common causes of events, such as explosions, gun shots etc.

2.1.2. Layered Architecture of SIRF

The proposed information representation framework take a layered architecture, with each layer represents a level of information abstraction. In general, four layers are defined, i.e. the signal layer, the feature layer, the primitive layer and the concept layer. The primitive layer is in the middle, which links lower layer sensor information with higher layer objective concepts. Figure 1 illustrates one example of the proposed representation framework. The component specifications and their linkages depend on the sensing system, the sensed environment and possible semantic queries.

At the bottom layer, sensors and sensor networks of various modalities produce raw sensor signals as noisy observations of the environment.

At the second layer, sensor data processing units extract various features that can be used in describing certain conceptual primitives, such as HOG (histogram of oriented gradient) for human objects, SIFT (scale invariant feature transform) for general visual objects, spectral features for acoustic event, and color histogram and statistical moments for background etc. Stochastic models such as HMM (hidden Markov model), GPDM (Gaussian process dynamic model) can also be considered as feature extraction tools for describing actions and processes, such as human motions or speech patterns.

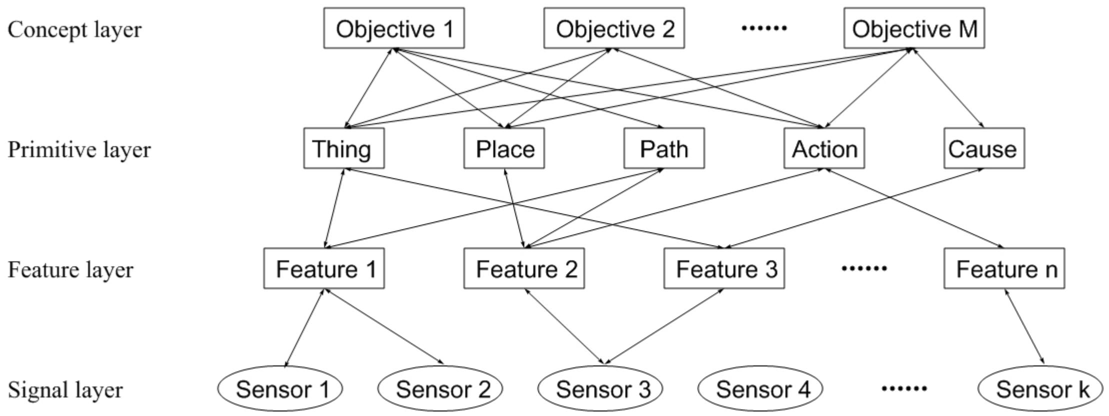


Figure 2.1. A layered information abstraction architecture in SSE systems.

At the third layer, conceptual primitives are constructed based on the descriptive features from the feature layer. Parametric or non-parametric classifiers on multi-modal feature sets are employed to affirm the existence of certain primitives and to determine the attributes of these primitives. The classification results can be described in likelihoods, which facilitate probabilistic inference of the queried concept.

At the top layer, multiple conceptual primitives form a semantic concept, or mission objective. A concept is not merely a joint event of these multiple primitives; instead it can be described in a form of a structured or graphical model consisting of these primitives. These models are referred to as SIRF concept models (SCMs). An example of such model is shown in Figure 2.1.

In general, layer partitioning is based on information abstraction. Sub-layers can be specified at the feature layer and the concept layer. The primitive layer can be connected to features and concepts at different sub-layers.

The query process takes the top-down path. For each query (or concept), an SCM can be constructed using the relevant conceptual primitives. These models can be defined by users based on domain knowledge. They can also be learned from training sensor data, through the process similar to association rule data mining. Given a construct of an SCM and the identified primitives, a set of features are formulates and a set of sensors are probed.

The inference process takes the bottom-up path. A set of sensor signals are first collected, and the relevant features are computed. Then the primitives are asserted, and the likelihood of the SCM is evaluated.

2.1.3. Implementation

SCMs can be easily described in XML, which makes this proposed semantic representation portable and interoperable. For example, “a human object walking through a security check point” can be described as:

```
<Place name= “a security check point”>
  <Thing name= “a human object” >
        <Path name= “trajectory going through the check point”><\Path>
        <Action name= “walking”><\Action>
  <\Thing>
<\Place>
```

We developed a simple parsing software, which can process video sequence and associated primitive information and generate an SIRF description in XML. Since the software is only to demonstrate the generation of SIRF in XML, we assume that the primitives such locations and actions of each object are already extracted and stored in an input XML file. The software is implemented in Python. It can be executed on Windows, Linux/Unix, Mac OS X etc.

The software consists of three components. The first component is to generate SIRF descriptions in XML; the second component is to perform a simple group merging operation; the third component handles user I/O and video display.

Graph structure is utilized to represent these objects in the video. Each node represents an object’s action in a fixed-length but varying-endpoint time interval. The root of the graph points to all objects in the first time interval. All the nodes of same object are linked in a path. Taking the advantage of Divide and Conquer principle, given n nodes in the video, the complexity of the algorithm is $O(n)$.

In the experiment, we evaluate the proposed algorithm on the BEHAVE dataset [25]. BEHAVE dataset consists of four video clips, with 76,800 frames in total. This video data set is recorded at 26 frames per second and has a resolution of 640×480. Group activities include *InGroup*, *Approach*, *WalkTogether*, *Split*, *Ignore*, *Following*, *Chase*, *Fight*, *RunTogether*, and *Meet*.

There are 6229 single nodes from the input. 220 nodes were merged, which means 220 groups were detected.

In the output XML, *Path*, *Place*, *Thing*, and *Action* are identified and labeled:

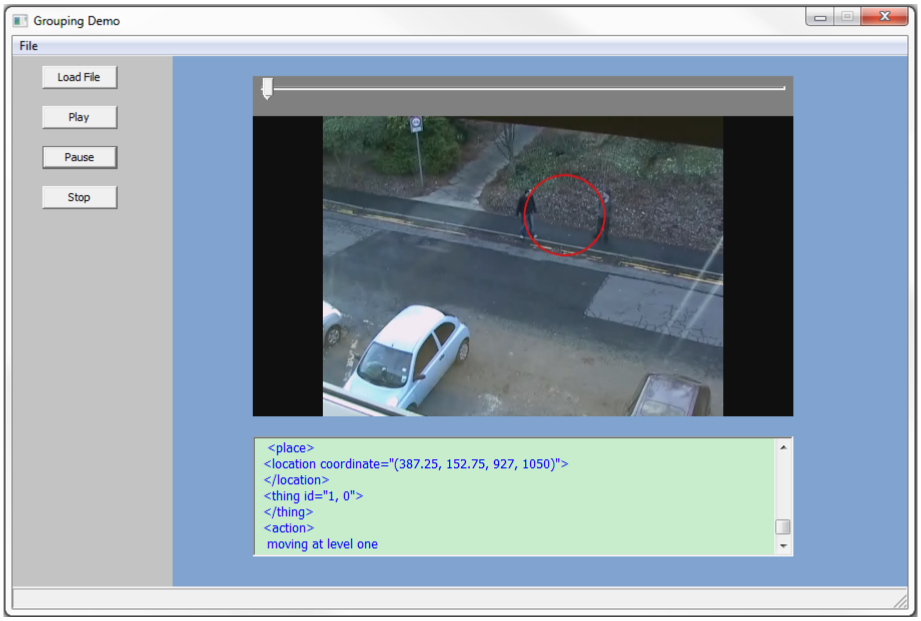
- Path consists of a sequence of places. Things and actions are moving along paths.
- Places are locations with some time duration. Things and actions occur in places.
- Things can be human or objects. They will conduct certain actions.
- Actions are dynamic behaviors of things, such as moving at speed level one/two/three

Here is a sample XML generate by this software:

```
<XML encoding="utf-8" version="1.0" />
<Path>
        <Place>
                <Location coordinate="(574.5, 288.5, 9139, 9139)" />
                <Thing id="3" />
                <Action>moving at level one</Action>
        </Place>
        <Place>
                <Location coordinate="(511.0, 316.5, 9465, 9465)" />
                <Thing id="4" />
                <Action>moving at level two</Action>
        </Place>
        <Place>
                <Location coordinate="(448.0, 267.5, 5851, 5852)" />
                <Thing id="3, 4" />
                <Action>moving at level two</Action>
        </Place>
</Path>
……
```

This segment of XML indicates that there are three *Places* under the *Path*. In the first *Place* element there is a person with id 3 moving at speed level one (less than 5 Pix per frame). Its location in the video frame is (574.5, 288.5) and it appears at frame 9139. In the second *Place* element there is a person with id 4 moving at speed level two (between 5 to 10 pix per frame). Its location in the video frame is (511.0, 316.5) and it appears at frame 9465. In the third *Place* there is a group which consists of objects 3 and 4. The location in the frame is (448.0, 267.5) and it appears between frame 5851 and 5852. The group is moving at speed level two (between 5 to 10 pix per frame)

A screen shot of this software is shown in Figure 2.2.



```
 <place>
<location coordinate="(387.25, 152.75, 927, 1050)">
</location>
<thing id="1, 0">
</thing>
<action>
 moving at level one
```

Figure 2.2. A screen shot of our SIRF demo software.

This software produces a first level description based on all primitive extractor outputs at a frame-by-frame basis. Subsequent primitive combination at syntactic level will be able to reduce the entries significantly. The resulting XML document will then be used in semantic level BN inference.

2.2. Learning of Syntactic Constructs in SIRF, a Probabilistic Context Free Grammar Approach

2.2.1. Introduction

At syntactic level, a new conceptual primitive can be iteratively constructed from a series of lower level primitives. This process is modeled through Probabilistic Context Free Grammar (PCFG) in SIRF.

Context free grammar is a commonly used in modeling constituency in natural language processing [9]. It is more flexible than regular grammars (equivalent to HMMs) in the Chomsky hierarchy [12]. Context-free grammar (CFG) can be expressed as G = (V, T, S, P), where V is a finite set of non-terminal characters or variables, T is a finite set of terminals characters disjoint from V, $S \in V$ is the start variable (or start symbol) representing the whole sentence, and P is a finite set of rewrite rules or productions of the grammar. The production rules are in the form of $X \rightarrow \lambda$, where $X \in V$ and $\lambda \in (V \cup T)$. As an extension to CFG, Probabilistic Context Free Grammar also defines a set of probabilities for every production rules Pr, and it is normalized for each l.h.s. symbol X in the production rule.

A critical distinction between linguistic construct and semantic sensing construct is the rich inter-relationships among multi-dimensional primitives. A critical distinction between linguistic construct and visual construct is the rich inter-relationships among multi-dimensional primitives. Several attempts have been made by other groups to introduce spatial and temporal relations among $(V \cup T)$. We take a more fundamental approach to this challenge, by encoding the relationships in a cognitive framework resembles human sensorimotor interactions. Basically, we describe low level primitives, T or V, as CL primitives, and we formulate their spatial temporal relationship matrices based on observations and our knowledge of these primitives. According to cognitive linguistic principles, conceptual primitives in SIRF inherently exhibit a set of production rules.

S → Path | Place
Path → Path Place | Place
Place → Thing Action | ε
Place → Path Path | Path
Action → Action Action
Thing → Thing Thing

These rules show that SIRF is capable of describing unlimited recursion of embedded symbols. Furthermore, additional production rules can be learned from data based on minimum description length (MDL) criteria. With these production rules, higher level primitives (i.e. phrases) can be constructed from lower level primitives (i.e. words) through “merge” and “construct” operations. Syntactic level abstraction is critical in describing semantics in multi-agent scenarios, because multiple *things* (e.g. human individuals) may form a new *thing* (e.g. a group), and individual *actions* (e.g. motions of body parts) may form a new *action* (e.g. group fighting). Given a set of production rules associated with different concepts, string parsing tools, such as Earley parser [24], can be used to parse unknown data sequences, and find the most likely production rule set to predict the concept of the sequence.

2.2.2. PCFG in Visual Sensing

Several recent studies on grammar models have shown promising potentials in visual event understanding and representation [13][14][15]. To apply grammar models for event recognition, usually low-level features are firstly extracted from videos and then classified into a set of terminal symbols, i.e. visual event primitives. Different event primitives will then form a discrete symbol string for syntactic analysis, including grammar induction and parsing. However, general linguistic models are essentially 1-dimensional sequential models. They are most suitable for sequential event recognition, regardless of the number of participants of the event. There are many scenarios with concurrent sub-events, such as a small group of people fighting each other, namely “group fighting”. The sub-event for each person in the group cannot be treated separately and sequentially. So the simple sequential approach has difficulties to represent such complex visual events.

To recognize parallel visual events, Joo et.al. [16] introduced attribute grammar for event recognition and anomaly detection. Recently, Zhang.et.al [17] extended probabilistic context free grammar to automatic learning of grammar rules and parallel parsing of sub-events simultaneously. Besides temporal semantics, spatial semantics have also been introduced to recognize two-person interactions [18]. These methods added attributes to each event primitive. For example, an ID set is stored in [17] and used for searching other concurrent event during parsing process. These approaches are suitable for a small number of parallel sub-events, and are very specific to certain scenarios, and can not be easily generalized.

We have introduced a cognitive linguistic (CL) based representation for visual events. Five different conceptual primitives, including place, path, action, thing and cause, are used to represent different visual events as shown in Figure 2.3. As this representation is derived from the fundamental constructor of human language, it can be intuitively applied to describe many kinds of complex visual events. Based on CL descriptions of visual events, we introduced a new method for small human group event parsing in video streams based on learned stochastic context free grammar models. Our method is able to describe spatial and temporal semantics for grammar induction. For the spatial semantics, both individual actions and small group behavior are taken into account for visual event representations. As to temporal semantics, the dynamic structures of multiple human objects in the scene are captured over time, which ensures a precise representation of all objects of interest.

As shown in Figure 2.3, a particular visual event is represented with five primitives. A “path” is composed of a sequence of places. A “place” is associated with the exact location and time duration a particular thing. “Things” can be human or objects, depending on different scenarios. “Action” is the corresponding action of the thing, which can be treated as visual primitive events in most aforementioned methods. A “cause” can be a special event or object causes other event or object to occur.

Unlike event-driven methods, the grammar learning is performed at two-level of thing-centric representations. In the first step, different things will be merged based upon their semantic distance. The merging process will be continued until it reaches the minimum semantic disorder [19], i.e. semantic social entropy. If multiple things have been merged as a group of things in the current place, a small group event recognition [20] will be performed and all human objects in the place will form a new thing, i.e. a "group", and its corresponding action will become the group action. If there are multiple groups, each group will be processed by the same procedure. If there are still individual persons outside the groups, they will maintain their individual descriptions.

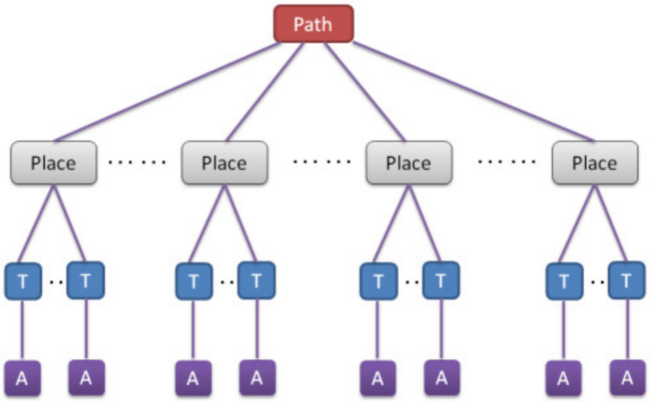


Figure 2.3. A CL based semantic representation of a visual event

After the merging of things, all the related concurrent events in the same place will be processed to form another high-level semantic representation, which avoids the parallel sub-event difficulty in the sequential grammar systems. The mixed descriptions of individual and group primitives will be used in the training of stochastic context free grammar rules. A minimum description length (MDL) based grammar induction method [22] has been used to the event sequence and different rules can be generated. The induced grammar rules will be used to parse different videos.

2.2.3. Proposed framework

Our proposed framework is focused on the small human group action recognition. Firstly, each human object's semantic information will be used for small human group detection. Once a small human group is discovered, their action will be recognized from group action classification. Otherwise each individual's action will be classified based on single human action recognition method.

Once these human action recognition results are obtained, a probabilistic context free grammar model will be initialized to automatically induce the potential rules behind group or individual action atoms. The induction will take into account both spatial and temporal correlation of each action and generate a number of rules to represent different combination of actions. These induced event rules then will be used to parse different testing videos.

Among the conceptual primitives in cognitive linguistics, ``path" provides an abstract description of activities, such as group meeting, group fighting, etc, over a certain space and time. ``Place" indicates the beginning and ending frame of the event, as well as the location of each agent in the group. ``Things" refers to group members, while ``action" is his/her activities in the ``place".

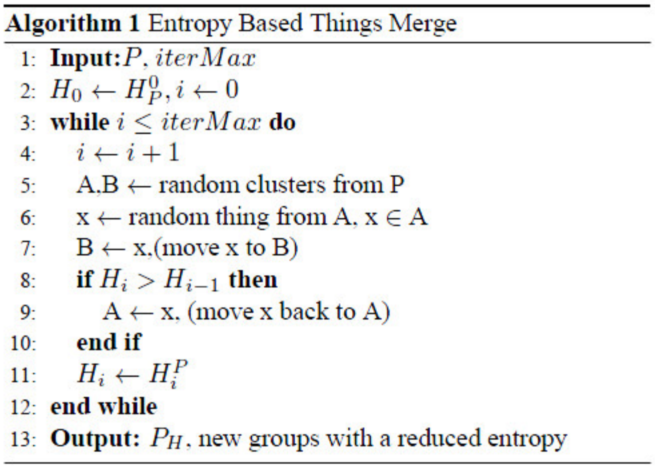
Given an input video with human object detection and tracking results, we divide it into small clips with one second each. Each clip is a ``place", and the all the human objects in the scene are ``things". Their movements are ``actions". Therefore we can construct a cognitive linguistic description of a video sequence in a 5-tuple representation. For example, {Walk Together, <x,y,w,h,t>, <person1, walking, person2, walking>} shows there are two people walking together.

2.2.4. Merge of Things

Inspired by several social metrics defined in social network analysis (SNA) [19], we define **semantic social entropy** for merging of things in the cognitive linguistic representation. Given a group *G* of *N* things, the semantic social entropy *H*(*G*) is defined as:

$$H(G) = -\sum_{i=1}^{N-1} \sum_{N}^{j=i+1} s_{ij} ln(s_{ij}) + (1 - s_{ij}) ln(1 - s_{ij}) \qquad (2.1)$$

where s_{ij} is the similarity measurement of two different things. In our case, each thing has a semantic description, which includes speed, direction, action, and location. s_{ij} is a measure of the semantic distance between two things. The procedure of "merging of things" is to maximize this social entropy when all the things are grouped into the correct cluster. The algorithm is described as the following.



Algorithm 1 Entropy Based Things Merge

```
Input: $P$, $iterMax$
$H_0 \leftarrow H_P^0, i \leftarrow 0$
while $i \leq iterMax$ do
    $i \leftarrow i + 1$
    A,B ← random clusters from P
    x ← random thing from A, $x \in A$
    B ← x,(move x to B)
    if $H_i > H_{i-1}$ then
        A ← x, (move x back to A)
    end if
    $H_i \leftarrow H_i^P$
end while
Output: $P_H$, new groups with a reduced entropy
```

The merging algorithm 1 requires an initial group partition. Here we used the minimum span tree to obtain the first group set based on their topological distribution. To evaluate our algorithm, we compare this algorithm with a recent group detection method proposed in [21] in Table 2.1.

Table 2.1. Human group detection performance

Data Set	Our Method	Method in [21]
SU1	60.4%	55.4%
SU2	71.5%	44.6%

[21] W. Ge, Collins R. T., and Ruback R. B., "Vision-based analysis of small groups in pedestrian crowds," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 5, pp. 1003–1016, 2012.

After successful merging of things, a desirable representation of the visual event takes the form as shown in Figure 2.4.

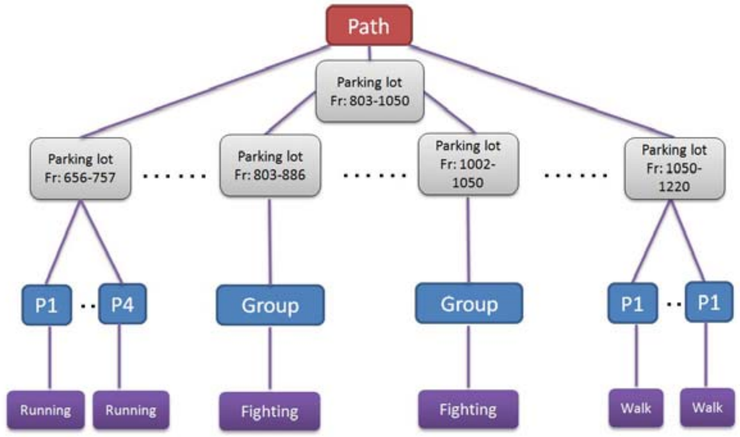


Figure 2.4. An example of syntactic parsing of a visual event

2.2.5. Hierarchical Grammar Rule Induction

Grammar rule induction has been studied for many decades. Minimum description length (MDL) [22] has been widely accepted as an effective criterion for grammar induction. As shown in natural language processing literature [23], grammar induction process iteratively performs the "merge" and "construct" operations on the training text, until it reaches the minimum description length. We use the following description length definition,

$$DL(t^L) = DL(t^L \mid G) + DL(G) \tag{2.2}$$

where t^L is the text sequence, **G** is the grammar, and *DL* is the description length. The "merge" and "construct" operations on the training of visual events can be described as the following.

- *Merge* : For each action pair (A,B), the merge operation will create a new candidate P → A|B.
- *Construct* : For each action pair A,B, the construct operation will produce a set of candidates, P → AB.

We adopted the basic procedure for visual events grammar induction. However, due to the complexity of visual events, we extended it to a semantic merge operation in our system. The basic merge operation is based on information theory, which compressed the event symbol sequence based upon entropy from signal process perspective. This principle is effective for grammar induction from text. However, as visual event primitives have much rich information besides of symbol itself. Some work has been done in this direction [17][18] from the spatial and temporal similarity of different events. Here we propose a semantic merge operation, which merge the things based on semantic representation. Take the group action "queuing" for an example, as shown in Figure 2.5., if there are eight people in a queue, the basic description are eight concurrent queuing action primitive, or eight *stand-move* events in a bottom-up description framework. As these are repetitive events, the basic merge operation will simply merge it to just one *stand-move* event, which will lead to a misrepresentation.

In our framework, we will apply a semantic merge operation of things as shown in the previous section. The basic idea is trying to find the most descriptive information which close to natural human language. In the "queuing" example, after semantic merge the system will output "a group of people is queuing" to describe such scene, which is more acceptable for human understanding. To do this, we firstly perform things merge at each place, if there is a human group, the group action will be classified based on our previously proposed human group action recognition method.

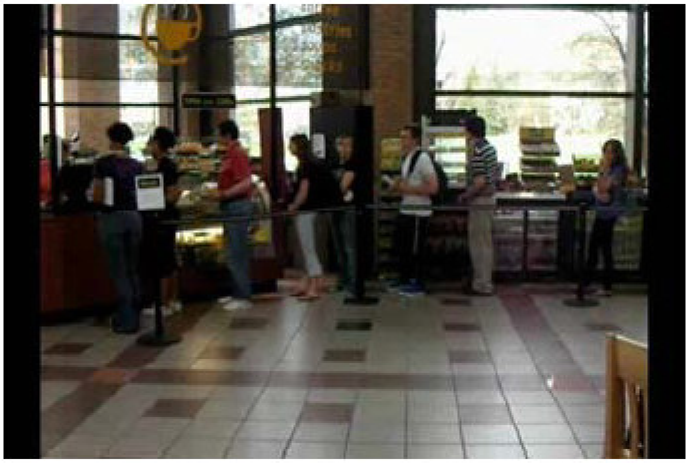
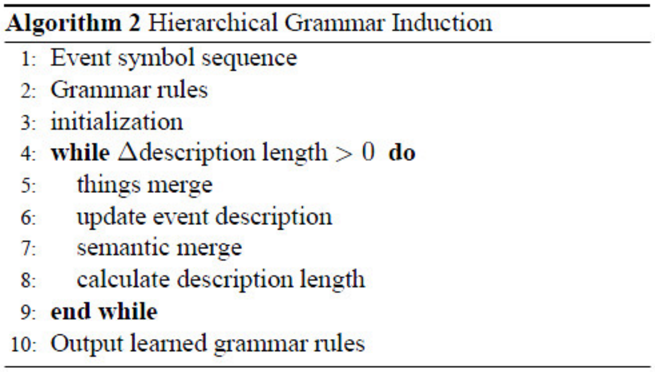


Figure 2.5. Action example: group queuing

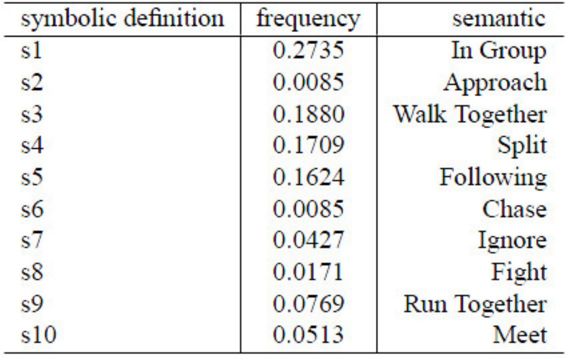


Algorithm 2 Hierarchical Grammar Induction

```
Event symbol sequence
Grammar rules
initialization
while $\Delta$description length $> 0$ do
    things merge
    update event description
    semantic merge
    calculate description length
end while
Output learned grammar rules
```

The procedure of grammar induction is shown in the Algorithm 2. As to the video event primitives, in this work, we define three different action primitive for individuals: “walk”, “run”, and “stand”. We also use ten action primitives for group actions from BEHAVE data set, including InGroup, Approach, WalkTogether, Split, Ignore, Following, Chase , Fight, RunTogether, and Meet, as shown from Table 2.2.

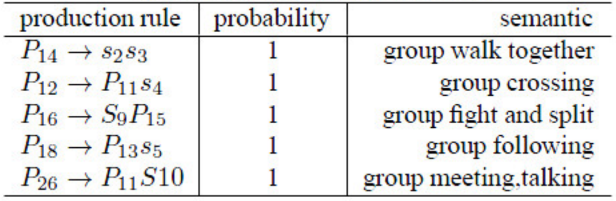
Table 2.2. Group Primitives



symbolic definition	frequency	semantic
s1	0.2735	In Group
s2	0.0085	Approach
s3	0.1880	Walk Together
s4	0.1709	Split
s5	0.1624	Following
s6	0.0085	Chase
s7	0.0427	Ignore
s8	0.0171	Fight
s9	0.0769	Run Together
s10	0.0513	Meet

Table 2.3 shows some examples of induced grammar rules, with highest probabilities, using the Algorithm 2, from the BEHAVE data set.

Table 2.3. Examples of learned rules



production rule	probability	semantic
$P_{14} \rightarrow s_2 s_3$	1	group walk together
$P_{12} \rightarrow P_{11} s_4$	1	group crossing
$P_{16} \rightarrow S_9 P_{15}$	1	group fight and split
$P_{18} \rightarrow P_{13} s_5$	1	group following
$P_{26} \rightarrow P_{11} S10$	1	group meeting,talking

An example of the learned visual phrase “group fighting” is shown in Figure 2.6.

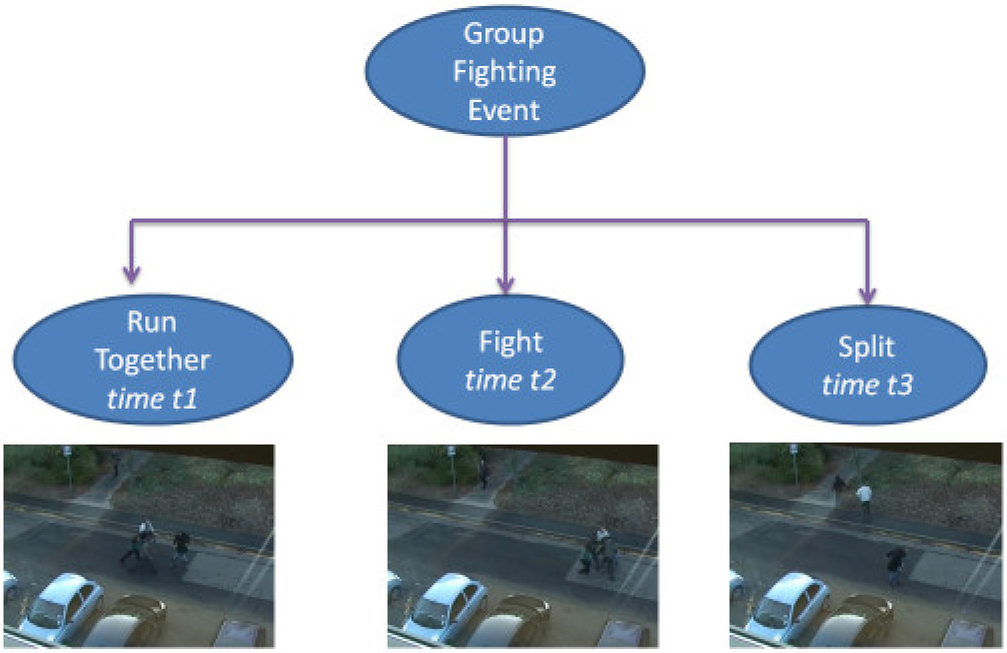


Figure 2.6. A visual phrase of “group fighting”.

2.2.6. Video Event Parsing

After extracting low-level features and performing classification, each individual’s action primitives will form a string for parsing. We utilize the Earley parser for parsing [24]. Each event is recognized based on the Maximum Likelihood criterion. The video event parsing can be iteratively processed through three steps: prediction, scanning and completing.

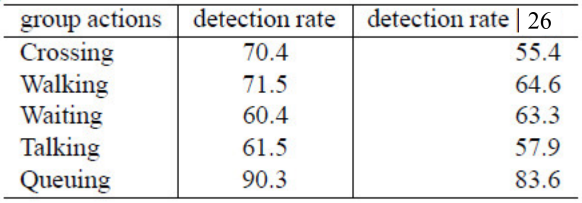
1. *Prediction* : A list of possible states will be generated based upon previous input.
2. *Scanning* : During scanning, the similarity between derived symbol and input string will be evaluated.
3. *Completing* : Based upon states selected from scanning step, the completing step will update all the positions for the pending derivations.

2.2.7. Experimental Results

In our experiments, we intentionally used two different datasets, one for training and the other for testing. The training, or ruler learning, was conducted on the BEHAVE dataset [25], as shown in the previous section. The testing, or parsing, was conducted on the Collective Activity dataset. This dataset contains 5 different collective activities, i.e. “crossing”, “walking”, “waiting”, “talking”, and “queuing”, in 44 short video sequences. Unlike BEHAVE data set, all the videos in this data set are recorded from real-world scenarios instead of controlled environment.

The recognition result is shown in Table 2.4. Compared to the benchmark results [26], our proposed method clearly demonstrated its advantage over the feature based methods. More importantly, the results show that the abstract grammar rules are applicable to different visual scenarios, which makes our method a viable solution to visual semantic information representation framework (SIRF).

Table 2.4. Recognition result on the Collective Activity data set



group actions	detection rate	detection rate | 26
Crossing	70.4	55.4
Walking	71.5	64.6
Waiting	60.4	63.3
Talking	61.5	57.9
Queuing	90.3	83.6

[26] W. Choi, K. Shahid, and S. Savarese, “What are they doing? : Collective activity classification using spatiotemporal relationship among people,” in IEEE 12th International Conference on Computer Vision Workshops, Kyoto, Japan, Oct. 2009, pp. 1282–1289.

2.3. Semantic Reasoning in SIRF, a Bayesian Network Approach

2.3.1. Bayesian Network in SIRF

While the words and simple phrases are parsed according to certain grammar models at the syntactic level, the resulting conceptual units (i.e. higher level conceptual primitives) can be organized together to form semantic concepts at the semantic level. Given the 5-dimensional conceptual primitives in our SIRF, a probabilistic semantic concept model can be naturally constructed through Bayesian networks [11].

A Bayesian network is defined by a directed acyclic graph (DAG) over nodes representing random variables and arcs signifying conditional dependencies between pairs of nodes. Let a set **X** = $\{X_1, ..., X_n\}$ of discrete variables where each variable X_n may take on values from a finite domain. A Bayesian network is a pair (*S*,**P**) where *S* is a network structure that encodes a set conditional independence assertions about variables in **X**, and **P** is a set of local probability distributions associated with each variable. That is, **P** = $\{Pi\}$ where $Pi = P(X_i|Pa_i)$, Pa_i denotes the parents of node *Xi* in *S*. The joint probability distribution for **X** can be expressed as

$$p(x) = \prod_{i=1}^{n} p(x_i \mid pa_i) \tag{2.3}$$

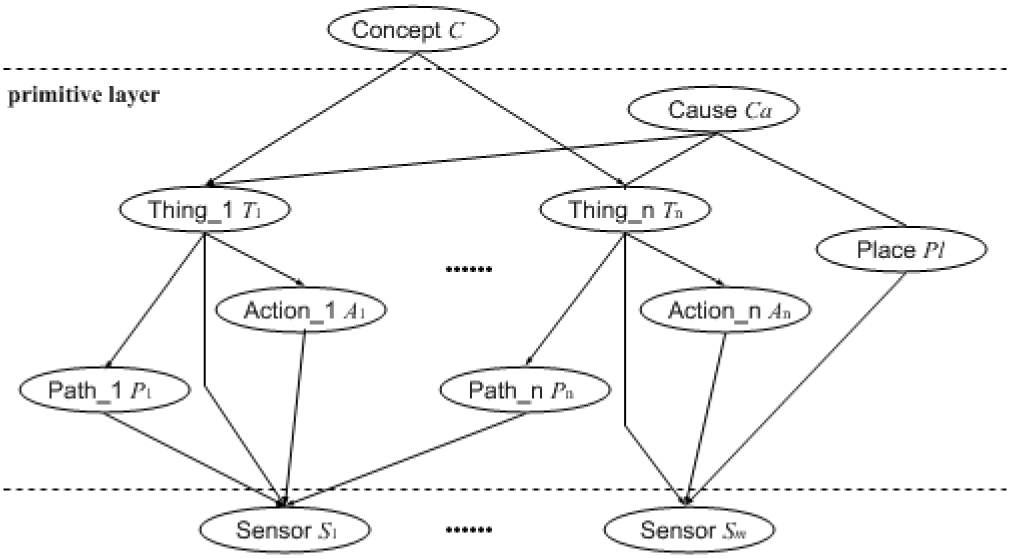


Figure 2.7. A BN SIRF concept model example

An example of a BN SCM is shown in Figure 2.7. We assume a concept *C* is a latent state. When the state *C* and cause *Ca* are true, a place (*Pl*) and a set of things ($T_{1...n}$) will be present. Each thing (T_i) will possess certain path (P_i) and actions (A_i). The existence of the place, things and their paths and actions will determine sensor observation (S_1 ... S_m), which represents the collective information from all accessible sensors and features.

2.3.2. Construction of BN for SIRF Concept Models

To construct a Bayesian network is generally not trivial. The structure of a BN usually encodes certain prior knowledge of experts. However, with cognitive linguistic conceptual primitives, it becomes intuitive for common user to encode causal semantics into BNs.

The initial step of this construction process is to collect all the relevant variables. In SIRF, these variables are conceptual primitives from the syntactic parsing. The conditional dependence between these variables will then be assessed from prior experience or from data. With this conditional dependence information, a directed acyclic graph *S* can be constructed.

For example, assume we are to describe a concept "hostile intent around a security check point during a specific day". This concept will be decomposed into the following SCM variables..

C represents the concept;
C_a represents possible external event or date;
Pl represents the area around the check point;
T represents types of human objects detected at the scene;
P represents types of motion trajectories of these human objects;
A represents types of the actions of these human objects;
$S_{1\ldots m}$ are the observation set consist of all calculated features.

Their conditional dependencies can be specified as the following.

$$\begin{aligned}
P(C|\Pi_C) &= P(C \mid Ca, Pl) \\
P(Ca|\Pi_{Ca}) &= P(Ca) \\
P(Pl|\Pi_{Pl}) &= P(Pl) \\
P(T|\Pi_T) &= P(T \mid Pl, C) \\
P(P|\Pi_P) &= P(P \mid T, C) \\
P(A|\Pi_A) &= P(A \mid T, C) \\
P(S_T|\Pi_{S_T}) &= P(S_T \mid T) \\
P(S_{Pl}|\Pi_{S_{Pl}}) &= P(S_{Pl} \mid Pl) \\
P(S_P|\Pi_{S_P}) &= P(S_P \mid P) \\
P(S_A|\Pi_{S_A}) &= P(S_A \mid A)
\end{aligned}$$

where Π_{X_i} is the subset of variables that X_i is conditionally dependent on.

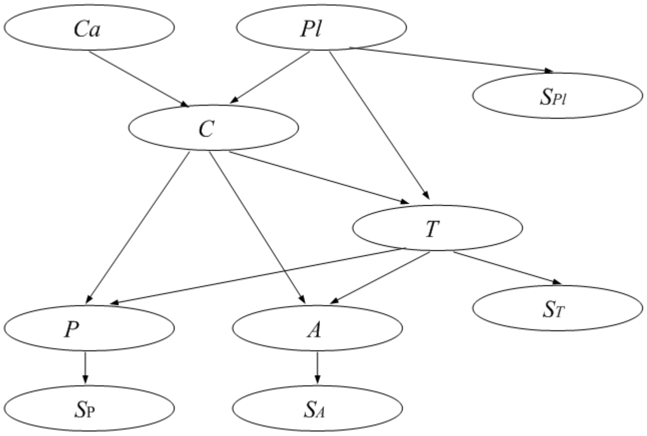


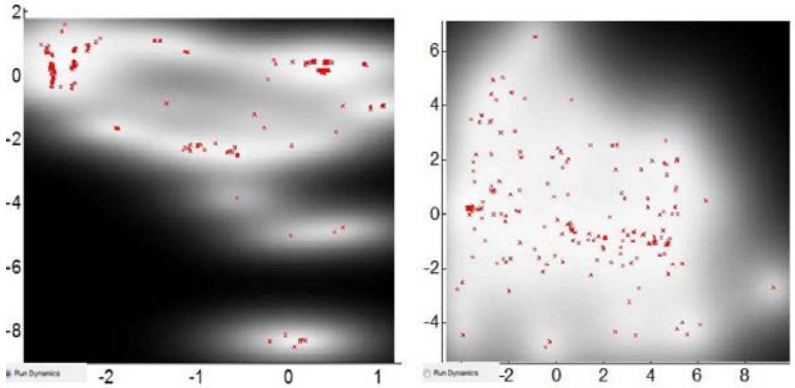
Figure 2.8. A BN SCM for "hostile intent around a security check point"

The resulting BN SCM is shown in Figure 2.8. As we can observe, this cognitive linguistics based BN construction process is intuitive and less ambiguous than general BN constructions.

2.3.3. Learning Local Probability Distributions

After the structure of the Bayesian network is determined, the next task is to learn all the local probability distributions $p(x_i \mid pa_i)$ from data. This can be achieved through various primitive modeling tools and methods, some of which will be introduced in the next section.

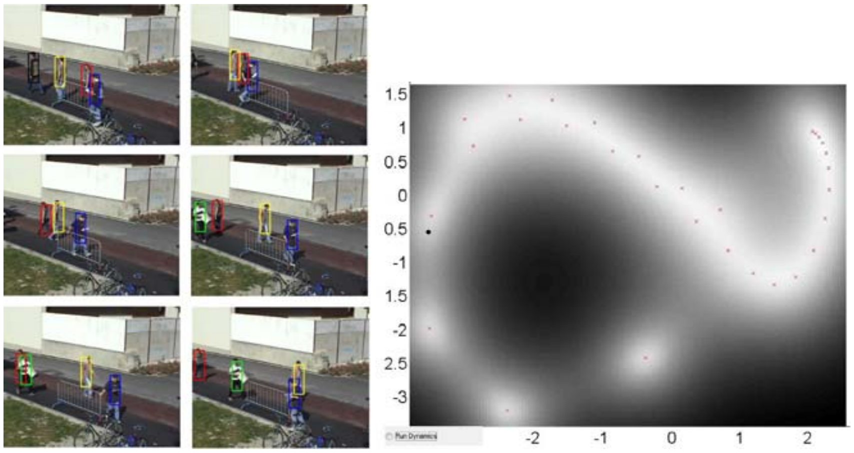
(a)



(b)

Figure 2.9. Examples of normal and abnormal *action* distributions

To demonstrate the feasibility of this BN-SCM method, Figure 2.9. and 2.10. show some examples of malicious action distribution $p(A_{\mathrm{M}}|A_{\mathrm{i}})$ and normal path trajectory distribution $p(P_{\mathrm{N}}|P_{\mathrm{i}})$. In Figure 2.9., group action types "In Group" and "Group Fighting", shown in Figure 2.9(a), are modeled using SN-GPDM, shown in Fig 2.9(b). In Figure 2.10., the trajectories of normal walking patterns, shown in Fig 2.10(a), are modeled using PF-GPDM, shown in Fig 2.10(a), so that the object with abnormal motion pattern, i.e. "walking in circle" pattern of the object marked by the blue box, can be identified as an outlier. The details of these modeling techniques will be discussed further in the following sections.



(a) (b)

Figure 2.10. Examples of normal walking *path* distribution

2.3.4. A BN SMC Use Case

In this subsection, we present a use-case example of BN SCM for "small human group behaviors" to demonstrate Bayesian reasoning with SCMs. A “small human group behavior” SCM is constructed as shown in Figure 2.11.

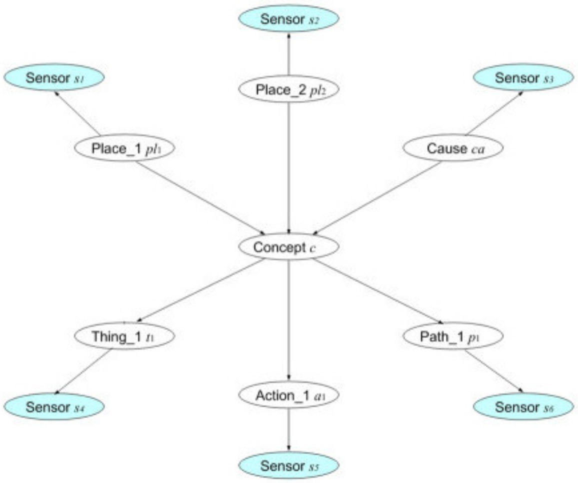


Figure 2.11. A BN SCM for "small human group behaviors"

In this graph, clear nodes are latent states which represent CL primitives; shaded nodes are observable nodes according to sensor modalities or features. Here we assume nodes are discrete variables, for illustration. In this construct, each observation node only assesses one primitive node, which simplifies the inference. The joint probability of the latent states can be expressed as

$$P(c , pl1 , pl2 , ca , t1 , a1 , p1)$$
$$= P(pl1) P(pl2) P(ca) P(c|pl1 , pl2 , ca) P(t1|c), P(a1|c) P(p1|c) \tag{2.4}$$

Assume that from observations or training dataset, the following priors and conditional probabilities are learned:

Place _1: (location)
pl1_1=location_1 (e.g. street)
pl1_2=location_2 (e.g. field)

P(pl1_1)	**P(pl1_2)**
0.4	0.6

Place_2: (time)
pl2_1=time_1 (e.g. morning)
pl2_2=time_2 (e.g. afternoon)
pl2_3=time_3 (e.g. evening)

P(pl2_1)	P(pl2_2)	P(pl2_3)
0.25	0.25	0.5

Cause:
ca_1=event_1 (e.g. special event)
ca_2=event_2 (e.g. no event)

P(ca_1)	P(ca_2)
0.01	0.99

Thing:
t_1=group_1 (e.g. small group)
t_2=group_2 (e.g. large group)

	P(t_1|c)	P(t_2|c)
c_1	0.4	0.6
c_2	0.6	0.4
c_3	0.5	0.5

Action:
a_1 = action atom_1
a_2 = action atom_2
a_3 = action atom_3

	P(a_1|c)	P(a_2|c)	P(a_3|c)
c_1	0.8	0.1	0.1
c_2	0.1	0.8	0.1
c_3	0.3	0.3	0.4

Path:
p_1 = path atom_1
p_2 = path atom_2
p_3 = path atom_3

	P(p_1|c)	P(p_2|c)	P(p_3|c)
c_1	0.1	0.8	0.1
c_2	0.8	0.1	0.1
c_3	0.3	0.3	0.4

Concept:
c_1= group sport
c_2= group fighting
c_3= other

	P(c_1|pl1,pl2,ca)	P(c_2|pl1,pl2,ca)	P(c_3|pl1,pl2,ca)
pl1_1,pl2_1,ca_1	0.1	0.3	0.6
pl1_1,pl2_1,ca_2	0.1	0.1	0.8
pl1_1,pl2_2,ca_1	0.2	0.4	0.4
pl1_1,pl2_2,ca_2	0.2	0.3	0.5
pl1_1,pl2_3,ca_1	0	0.5	0.5
pl1_1,pl2_3,ca_2	0	0.1	0.9
pl1_2,pl2_1,ca_1	0.8	0.1	0.1
pl1_2,pl2_1,ca_2	0.4	0.1	0.5
pl1_2,pl2_2,ca_1	0.8	0.2	0
pl1_2,pl2_2,ca_2	0.6	0.2	0.2
pl1_2,pl2_3,ca_1	0.8	0.1	0.1
pl1_2,pl2_3,ca_2	0.2	0.2	0.6

Once such knowledge is obtained, reasoning through SRIF can be made. For example, if full knowledge is available, the following probabilities can be calculated:

IF: p1_1 (street), p2_2(afternoon), ca_2(no event), c_2(group fighting), t_2(small group), a_ 1(action atom1), p_1(path atom1)
THEN: Joint probability P=0.00144

IF: p1_1 (street), p2_2(afternoon), ca_2(no event), c_2(group fighting), t_2(small group), a_ 2(action atom2), p_1(path atom1)
THEN: Joint probability P=0.01152

IF: p1_1 (street), p2_2(afternoon), ca_2(no event), c_2(group fighting), t_2(small group), a_ 3(action atom3), p_1(path atom1)
THEN: Joint probability P=0.00144

Very frequently some of the probabilities are not available. SIRF can well handle such circumstances through marginalization. For example, if there is no input for cause, path and thing, the following probabilities can be calculated:

IF: p1_1 (street), p2_2(afternoon), a_ 1(action atom2)
THEN: conditional probability for c_1(group sport) is P(c_1|p1_1,p2_1,a_2)=0.0644

IF: p1_1 (street), p2_2(afternoon), a_ 2(action atom2)
THEN: conditional probability for c_2(group fighting) is P(c_2|p1_1,p2_1,a_2)=0.7750

IF: p1_1 (street), p2_2(afternoon), a_ 3(action atom2)
THEN: conditional probability for c_3(other) is P(c_3|p1_1,p2_1,a_2)=0.1606

In this case the prediction would be "group fighting".

SIRF is designed for rich semantics with probabilistic inference capability. Compared with the common alternative of semantic representation, e.g. ontology, SIRF has clear advantages in sensing and surveillance applications. However it shares typical challenges for Bayesian Networks. We are further extension of this framework to incorporate linguistic modeling and un-supervised learning capabilities.

3. Primitive Modeling in SIRF

3.1. Dynamic Structure Preserving Map (DSPM) Method for Individual Human Action Primitive Modeling

3.1.1. Introduction

Human action recognition has attracted much attention in the fields of computer vision and machine learning in recent years [27]. Many previous works have focused on augmenting the feature descriptions, such as proposing stronger feature sets and combining different features [28], or improving action recognition models, such as clustering and classification for scene analysis or abnormal events detection [29]. The analysis of human actions in a video sequence is challenging, because the recognition system is required to extract implicit properties including spatio-temporal coherence, behavior dynamics, and shape deformation. The action feature extraction from a video sequence is different from static image analysis, since spatio-temporal variation might result in meaningful behavior patterns. For example, the changes in human motion orientation or gesture during a specified time interval may indicate what actions might have occurred. In real world applications, irregular behaviors or environments should also be taken into account, which requires the dynamic model to adapt to unexpected factors.

In this work we introduced a dynamic structure preserving map (DSPM) for action clustering and recognition, with emphasis on unsupervised clustering. DSPM is an extension to self organizing map (SOM) in capturing spatial-temporal dependency in video sequences. DSPM has several unique properties.

1) DSPM is able to learn low-level features and produce a generative model to represent the dynamic topological structure. Instead of extracting carefully selected features, our method can automatically learn intrinsic characteristics from raw optical flow field for action recognition. Extending to the conventional SOM models, DSPM accumulates dynamic behavior of best-matching units (BMUs) to adjust their synaptic neuron weights, which can effectively capture the temporal information.

2) DSPM can aggregate the spatial-temporal clustering while simultaneously preserve underlying topological structure. Characterized by the parameters of latent neural distribution and neighborhood kernel function, the highly relevant spatial-temporal correlations for each action feature set are adaptively preserved in a 2-D lattice of neurons.

3) DSPM provides an effective way to reduce the dimensionality of input raw feature set, such as dense optical flow, to represent human motions in videos. Through the non-linear mapping procedure, DSPM can reduce the computational cost and data redundancy in action recognition.

The DSPM method contains a series of operations. Fields of optical flow are first calculated from consecutive video frames. Each field vector is mapped to one neuron in the DSPM according to competitive and adaptive learning rules (Algorithm 1). An adaptive neuron merging scheme is applied for cluster optimization (Algorithm 2). Based on the clusters on the spatial-temporal feature map defined in DSPM, the parameters of the latent space Markov model are estimated. The ensemble learning based on EM further enhances the dynamic model to yield better performance. The classifier with highest likelihood will be selected to predict class label. Normally there are significant amount of redundancy in action video sequences, especially at the beginning and the end of the sequence. A frame down sampling and simple motion based frame selection is applied in the pre-processing stage. The basic learning procedure of DSPM is illustrated in Fig. 3.1.

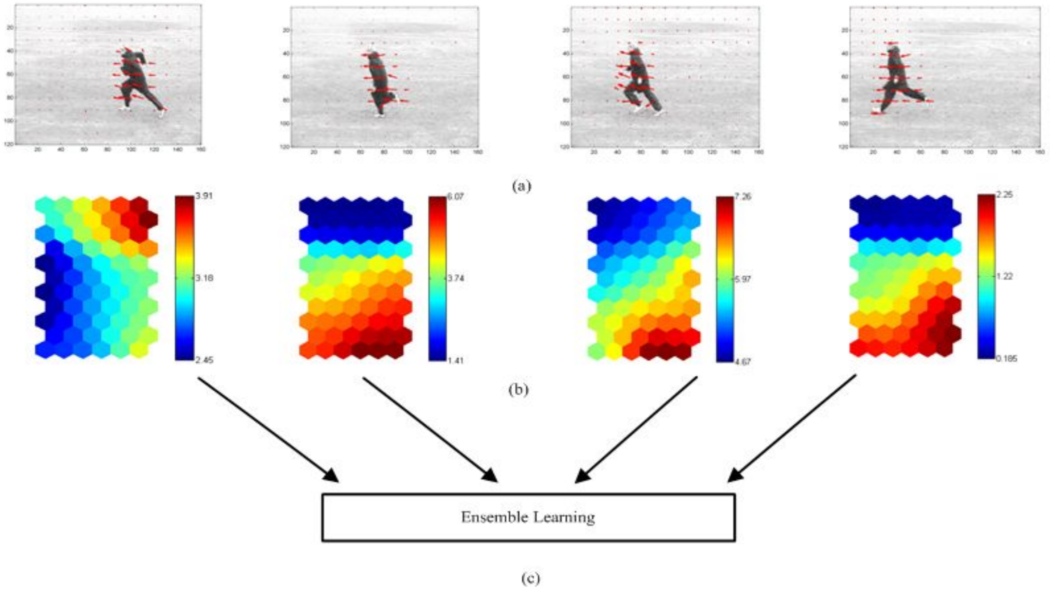


Figure 3.1. DSPM learning process. (a) Optical flow is extracted from each action video sequences. Given two consecutive frames, optical flow is computed at each pixel, and sampled with a 10×10 grid. For instance, the frame size of KTH data set is 160×120, after optical flow computing, the size of optical flow field for each frame is 16×12×2. The third dimension 2 indicates the magnitude and direction of optical flow. (b) Example DSPMs describing spatio-temporal patterns. The colors of grid represent the distances of various motions on DSPM. (c) The EM based ensemble learning is adopted to predict the action class.

3.1.2. Related Works

As a typical classification problem, feature extraction plays an essential role in the action recognition. Due to the intrinsic sequential property, many spatio-temporal features, such as STV [31], STIP [32][33], HOSVD [34] have been developed. Besides the spatio-temporal property, feature sets with multiple hierarchies are also introduced for action recognition. Sun et al. [35] modeled spatio-temporal context information in a hierarchical structure. Three levels of context were established in ascending order of abstraction: point-level context, intra-trajectory context, and inter-trajectory context. Gilbert et al. [36] introduced a novel approach to use very dense corner features, which were spatially and temporally grouped in a hierarchical process to produce an overcomplete compound feature set. In addition, the spatio-temporal feature set is also combined with other features, such as shapes [37], to make the action more descriptive.

Besides augmenting the features, different machine learning algorithms also have been introduced to improve the human action recognition performance. Zhu et al. [38] adopted multi-class support vector machine (SVM) with linear kernels. Schuldt et al. [39] used local space-time features for recognizing complex motion patterns. They constructed video representations in terms of local space-time features and integrated these representations with SVM classification schemes for action recognition. To improve the robustness, a Multiple Kernel Learning with Augmented Features (AFMKL) was proposed to learn an adapted classifier based on multiple kernels and pre-learned classifiers of other action classes in [40]. Fathi et al. [41] classified the input video sequence into one of the discrete action classes. The low-level motion features were used as the weak classifiers. The mid-level shape features were constructed from low-level gradient features using AdaBoost. To aggregate the information from different parts of the video sequence, AdaBoost was used for a second time to train the final classifier from the mid-level motion features.

Rather than computing the hand-engineered features or introducing complex classification models, we adopted the feature learning concept in [28] and [42], together with SOM model, to build DSPM to persevere the underlying highly relevant structure both in spatial and temporal dimension.

3.1.3. Dynamic Structure Preserving Map for Action Recognition

It is a complex process to analyze the correlation and variation across space and time. There are limitations on the estimation of traditional state-space models, since the high dimensional parameters may lead to complex dependency structures. Based on the clusters on the spatio-temporal feature map defined in DSPM, the parameters of the latent space model are estimated. The ensemble learning based on EM further enhances the dynamic model to yield better performance. The classifier with highest likelihood will be selected to predict class label. The training procedure of DSPM can be illustrated in Figure 3.1.

Self-organizing Map

SOM [30] is considered as a powerful neural network model in unsupervised learning, which can extract certain implicit knowledge without human intervention or empirical evidence. Given the input data sequence $X = \{x_1, ..., x_n\}$ and synaptic neuron weight m_j, $j \in \{1, ..., N_s\}$, N_s is the total number of the neurons on the map. The procedure of searching the best-matching unit (BMU) can be expressed as Eq. (3.1).

$$b_i = \arg\min_j || x_i - m_j || \tag{3.1}$$

Gaussian neighborhood kernel function defined in Eq. (3.2) is used to constrain the neighborhood scope of the BMU.

$$h_{j,b_i}(t) = \exp(-\frac{d_{j,b_i}^2}{2\sigma^2(t)}) \tag{3.2}$$

where $d_{j,b_i} = \| r_j - r_{b_i} \|$, $\sigma(t) = \sigma_0 (\frac{\sigma_1}{\sigma_0})^{\frac{t}{N_c}}$, r_j is a 2-D position vector of neuron j; t represents the training time; N_c denotes the convergence iterations; σ_0 and σ_1 are initial and terminal neighborhood radius, respectively.

An adaptive learning rule updates the synaptic neuron weight $m_{j,t+1}$ according to Eq. (3.3).

$$m_{j,t+1} = m_{j,t} + \alpha(t) h_{j,b_i}(t)(x_i - m_{j,t}) \tag{3.3}$$

where $\alpha(t) = \alpha_0 (\frac{\alpha_1}{\alpha_0})^{\frac{t}{N_c}}$, α_0 and α_1 represent the initial and terminal learning rate, respectively.

Based on the learning procedure, an elastic map is formed for the input data, as shown in Figure 3.2. The clustering procedure indicates that synaptic neurons are more likely to move towards the dense area.

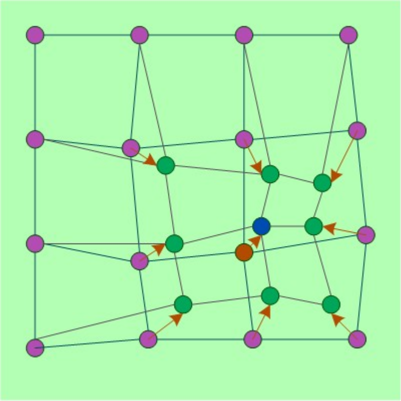


Figure 3.2. Illustration of adaptive learning in SOM.

Dynamic Structure Preserving Map

In SOM, the neighborhood function can be only used to preserve the spatial topology. Several extensions to SOM, including Temporal Kohonen map (TKM) and recurrent self-organizing map

(RSOM) [43], have been proposed to adaptively model a data distribution over time on non-stationary input sequences. Although TKM preserves a trace of the past activation in terms of weighted sum, the weights are only updated towards the last frame sample of the input sequence based on the convention SOM update rule. RSOM provides a consistent update rule for the network parameters. The main objective of TKM and RSOM is to follow the trend of the temporal sequence while smoothing out temporary volatilities. These methods emphasize more on the latest samples, and eventually remove the influence from old samples. On the contrary, DSPM intends to capture the whole dynamic patterns within the data sequence. We improve the learning rule of DSPM based on Eq. (3.3). The input sequential samples, after some simple cleaning operation, have the same importance and contribute evenly to the model from the beginning to the end. The resulting DSPM with the complete spatio-temporal information is then used in classification. In particular, the neuron transition probabilities in DSPM can describe the temporal dynamics from the training video sequences. DSPM models sequential dynamics by introducing Markov process to capture neuron transition probabilities between every two time samples. It is similar to Markov random walk [44] on graph, where at each step the walk jumps from one place to another based on specified probability distribution. The parameters of Markov process are used in neuron update and model classification.

At each frame of a feature sequence, this algorithm uses the conventional SOM competitive learning to find a best match unit/neuron (BMU) $b_{i,t}$ according to minimum l_2 distance. Then a unique weighting function is calculated to update the neuron vector in the SOM adaptive learning stage. The weighting function is defined as

$$p_{i,j} = \frac{K(i,j)}{\sum_{m \in N_s} K(i,m)} \tag{3.4}$$

which measures the distance between the previous neuron location and the current neuron location of the feature sequence. The kernel function K is: $K(i,j) = exp(-d(i,j)/\alpha)$, where $d(i,j)$ is Manhattan distance between BMU i at the time t to BMU j at the time $t+1$ on the lattice map, α is a constant.

Figure 3.3. illustrates the adaptive learning rule of DSPM. $m^*_{j,t+1}$ and $m_{j,t}$ are used to update the synaptic weights. We can see that $m^*_{j,t+1}$ can be calculated by $m_{j,t}$ and $m_{j,t+1}$. $m_{j,t}$ means the neuron weight at the previous time. The transition probabilities constrain the variations of neuron weights, which keeps temporal dependencies between $m_{j,t}$ and $m_{j,t+1}$. This formulation means the elastic characteristics of DSPM have effects on both spatial domain as well as temporal domain. The temporal properties depend on neighborhood topology and dynamic information. The synaptic neuron vector will then be updated according to

$$\begin{aligned} &m_j(t+1) = m_j(t) + \alpha(t) h_{j,b(x_i(t))}(t)(x_i(t) - p_{b(x_i(t)),b(x_i(t+1))} m_j^*(t+1) - (1 - p_{b(x_i(t)),b(x_i(t+1))}) m_j(t)) \\ &where \quad m_j^*(t+1) = (1 - p_{b(x_i(t)),b(x_i(t+1))}) m_j(t) + p_{b(x_i(t)),b(x_i(t+1))} m_j(t+1) \end{aligned} \tag{3.5}$$

This learning scheme distinguishes DSPM from all other SOM extensions, including TKM and RSOM, which are popular modifications with temporal elements.

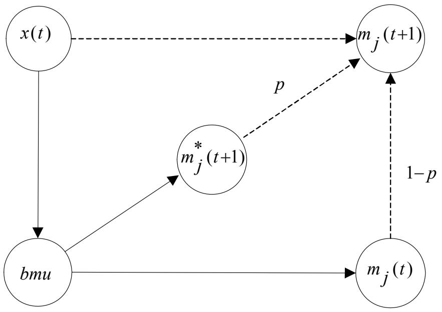
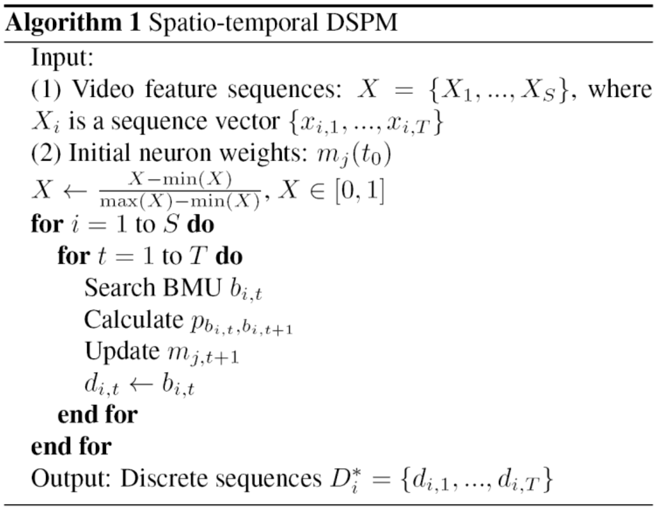


Figure 3.3. Adaptive learning rule of DSPM

The baseline DSPM learning algorithm can be described as follows.



```
Algorithm 1 Spatio-temporal DSPM
  Input:
  (1) Video feature sequences: $X = \{X_1, ..., X_S\}$, where
  $X_i$ is a sequence vector $\{x_{i,1}, ..., x_{i,T}\}$
  (2) Initial neuron weights: $m_j(t_0)$
  $X \leftarrow \frac{X-\min(X)}{\max(X)-\min(X)}$, $X \in [0, 1]$
  for $i = 1$ to $S$ do
    for $t = 1$ to $T$ do
      Search BMU $b_{i,t}$
      Calculate $p_{b_{i,t},b_{i,t+1}}$
      Update $m_{j,t+1}$
      $d_{i,t} \leftarrow b_{i,t}$
    end for
  end for
  Output: Discrete sequences $D_i^* = \{d_{i,1}, ..., d_{i,T}\}$
```

We take the frame samples of ``bend" action from 9 persons in the Weizmann dataset in Figure 3.4. DSPM can extract the key feature information by spatio-temporal knowledge and statistically measure the dependency by Markov transition probability. The green color grid is the output of DSPM, which can aggregate the key features into the clustering. The *x* coordinate represents temporal feature in frame number and the *y* coordinate means the cost on distance between the input video frame and its best matching neuron in DSPM. The red marked circles represent the corresponding cluster in the DSPM. We can see that key features with sparse distribution have a high cost on distance.

The interesting region of key features is near the arms or legs. When two arms or legs extend, the distance increases.

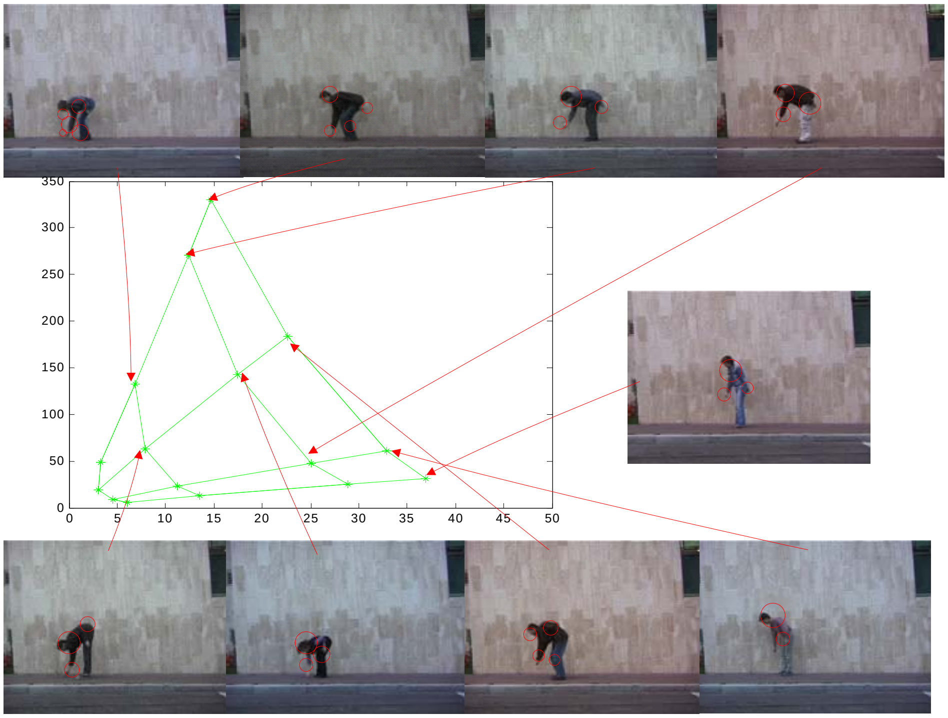


Figure 3.4 Spatio-temporal dependency analysis on key regions.

For “bend” action, the distance reaches the maximum value when the frame number is 23. For “walk” action, there are several peaks when the frame numbers are 6, 15, 25, 35, 45, as shown in Figure 3.5.

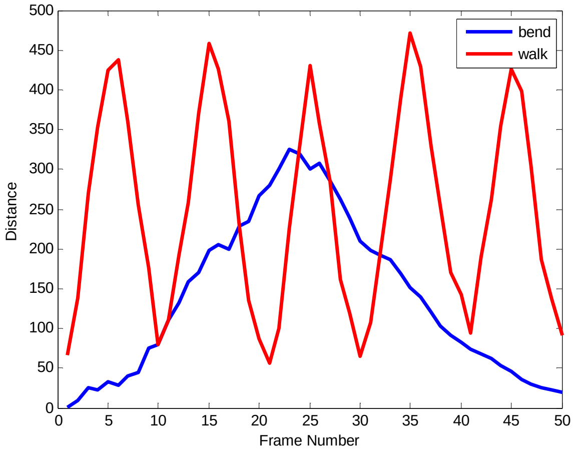


Figure 3.5. Distance between the input vector and its best matching neuron in “walk” and “bend” sequences.

Convergence Analysis

The form of Eq. (3.5) can be rewritten as Eq. (3.6). It is obvious that the denominator in Eq. (3.6) reflects the effects of spatio-temporal dependency in the learning procedure, which is crucial for extracting the motion behavior patterns in video sequences.

$$m_j(t+1) = m_j(t) + \frac{x(t) - m_j(t)}{\frac{1}{\alpha(t) h_{j,b(x_i(t))}(t)} + p^2_{b(x_i(t)),b(x_i(t+1))}} \tag{3.6}$$

Now we can verify the convergence of DSPM. As $t \to +\infty$, the learning rate $\alpha(t) \to 0$, $h_{j,b_{i,t}}(t) \to 0$, then

$$\frac{1}{\alpha(t) h_{j,b_{i,t}}(t)} \to +\infty \tag{3.7}$$

Since $0 < p_{b(x_i(t)),b(x_i(t+1))} < 1,$ we can obtain

$$\frac{1}{\alpha(t) h_{j,b(x_i(t))}(t)} + p^2_{b(x_i(t)),b(x_i(t+1))} \to +\infty \tag{3.8}$$

Therefore $\Delta m_j = m_j(t+1) - m_j(t) \to 0$.

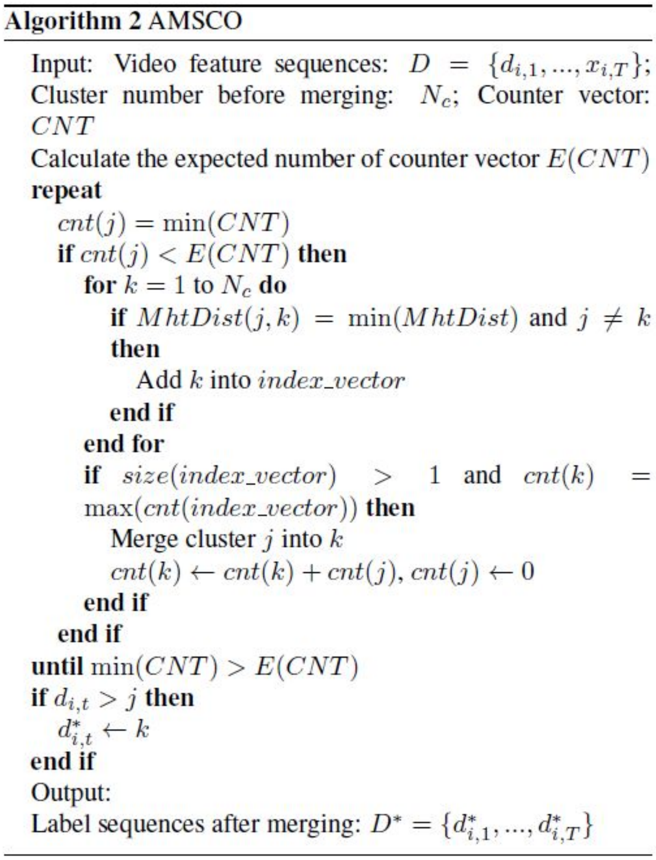
This proves the convergence of DSPM training procedure. The convergence rate is a key factor of learning time. Since $0 < p_{b(x_i(t)),b(x_i(t+1))} < 1,$, it will take more time in the initial training phase, and gradually reduce to 0. In the initial phase, the kernel function *K* in Eq. (3.4) specifies more neighbors in the initial scope, so more computation time is needed.

3.1.4. Adaptive Merging Strategy for Clustering Optimization

DSPM have a capacity to handle the high-dimensional problems through non-linear mapping. Different from other dimension reduction techniques, DSPM can preserve hidden useful spatio-temporal information in topological structure. The input sequential data are projected to the corresponding best-matching neurons. Finally, the outputs of DSPM will be discrete sequence data rather than high-dimensional data. The discrete sequence pattern will significantly reduce the computation cost and data redundancy in the training procedure in dynamic model. However, due to the characteristics of DSPM, the abnormal data or noises might push the minority of synaptic neurons move away to sparse data space. Data cleaning is an alternative tool to diminish this negative influence. But it will destroy the integrity and originality of the experimental data. Hence an adaptive merging strategy is presented to optimize the clustering by DSPM. The neighborhood relation is an important criterion to evaluate the similarities between data. However, the neighborhood solution probably results in local optimum. The local optimum can produce useful information by the adaptive merging strategy to approximate the global optimum.

After careful analysis on large amount of DSPM results, we realized that the neuron mapping, which is essentially a clustering operation, can be further optimized to increase sparsity of the neuron sequences. We introduced an adaptive merging scheme for clustering optimization (AMSCO) to analyze these sparse clusters and then merge them into potential clusters associated with spatial-temporal dependencies. The advantage of AMSCO is to avoid the local optima caused by DSPM. This adaptive clustering and cluster merging method creates a robust latent space to facilitate dynamic modeling over a temporal sequence. The spatial and temporal topology knowledge is used to analyze the adaptive merging strategy. The Manhattan distance is a key metric to evaluate spatiotemporal relationships. min(*CNT*) is a non-zero constant, which means the clusters in the sparse feature space. The purpose of AMSCO is to analyze these sparse clusters and then merge them into potential clusters associated with spatiotemporal dependencies. The adaptive clustering method creates a robust latent space to establish dynamic modeling framework of dynamic model.

The AMSCO algorithm can be described as follows.



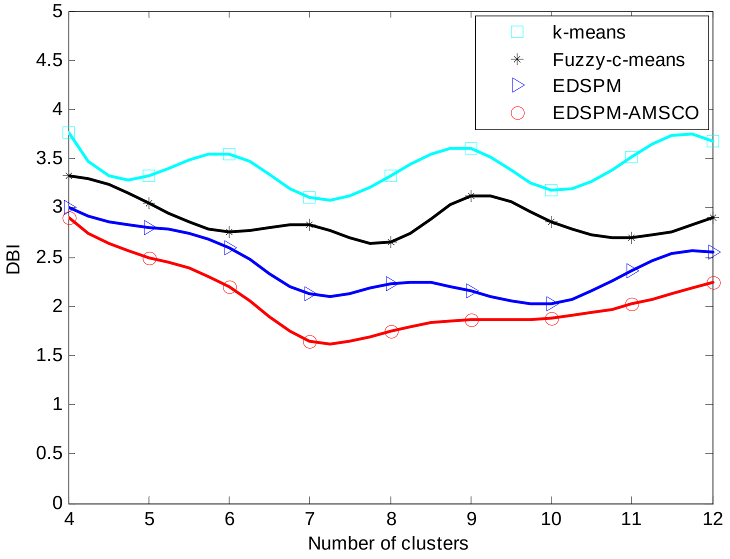
Algorithm 2 AMSCO

```
Input: Video feature sequences: $D = \{d_{i,1}, ..., x_{i,T}\}$;
Cluster number before merging: $N_c$; Counter vector: $CNT$
Calculate the expected number of counter vector $E(CNT)$
repeat
  $cnt(j) = \min(CNT)$
  if $cnt(j) < E(CNT)$ then
    for $k = 1$ to $N_c$ do
      if $MhtDist(j,k) = \min(MhtDist)$ and $j \neq k$ then
        Add $k$ into $index\_vector$
      end if
    end for
    if $size(index\_vector) > 1$ and $cnt(k) = \max(cnt(index\_vector))$ then
      Merge cluster $j$ into $k$
      $cnt(k) \leftarrow cnt(k) + cnt(j)$, $cnt(j) \leftarrow 0$
    end if
  end if
until $\min(CNT) > E(CNT)$
if $d_{i,t} > j$ then
  $d^*_{i,t} \leftarrow k$
end if
Output:
Label sequences after merging: $D^* = \{d^*_{i,1}, ..., d^*_{i,T}\}$
```

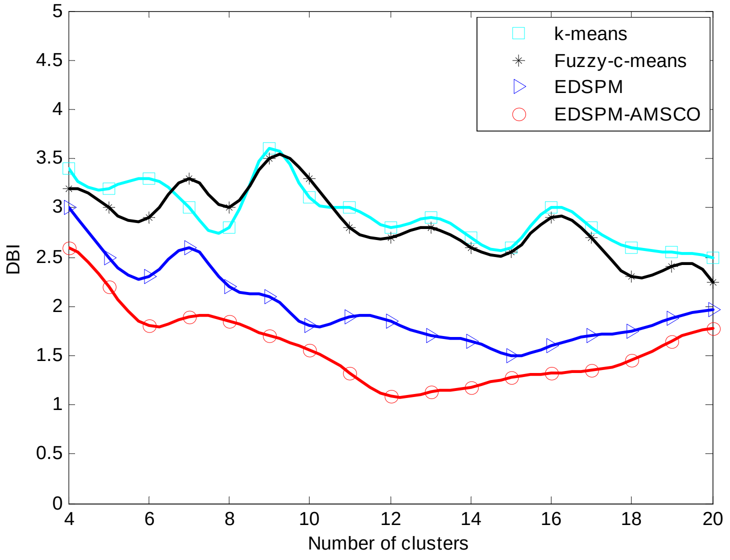
To analyze the effectiveness of the AMSCO algorithm in unsupervised clustering, we calculate the Davies–Bouldin index (DBI) to evaluate its clustering performance.

$$DB = \frac{1}{N}\sum_{i=1}^{N}\max_{j\neq i}\left(\frac{S_i + S_j}{dist(c_i, c_j)}\right) \tag{3.10}$$

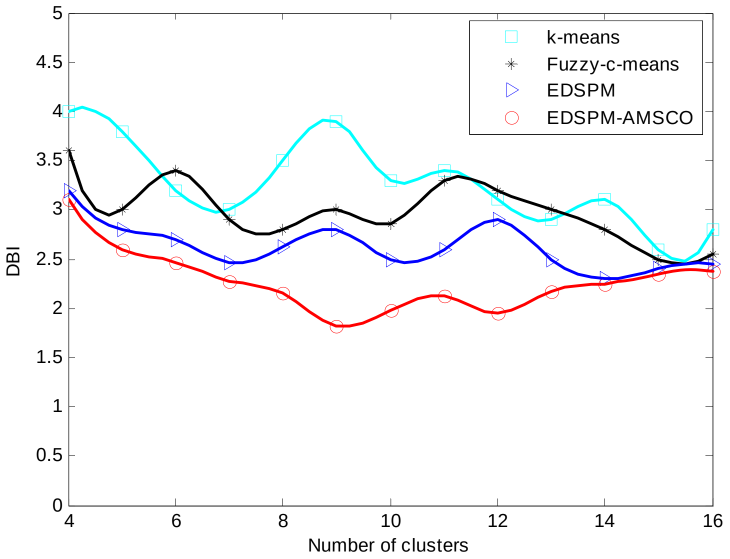
where N is the total number of clusters, c_i means the centroid of cluster i, $dist(c_i,c_j)$ is the inter-distance between centroids c_i and c_j, S_i is the average intra-distance within cluster i. In general, lower DBI indicates better data clustering.



(a)



(b)



(c)

Figure 3.6. Clustering performance on various datasets. (a) KTH, (b) Weizmann, (c) UCF.

We compare our methods with two conventional clustering methods, i.e. k-means and fuzzy-c-means. As shown in Figure 3.6., both k-means and fuzzy-c-means perform similarly, and these two popular clustering methods achieve lower performance than DSPM. DSPM with AMSCO obtains even better clustering results than DSPM. The variation of DBI in DSPM-AMSCO also appears smaller.

This study shows that DSPM-AMSCO can achieve minimum and meaningful clustering result in an unsupervised fashion. It is significant to our SIRF because unsupervised learning tools can help to extract semantic primitives in unknown scenarios.

After DSPM clustering and AMSCO optimization, a sequence of action feature vectors can be mapped into a sequence of DSPM neurons. Several examples of the neuron sequences for the actions in KTH dataset and Weizmann dataset are shown in Figures 3.7 and 3.8. These neuron sequences will be modeled the Markov random walk, and the parameters will be learned from training data through the EM algorithm for each action type.

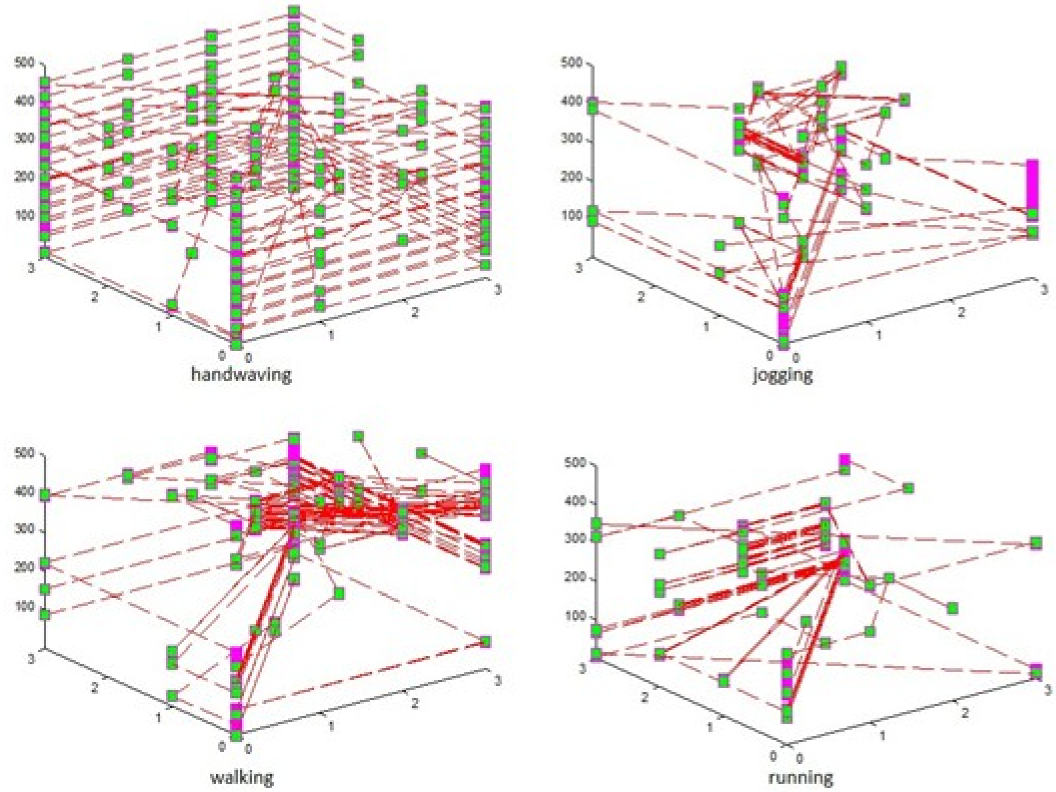


Figure 3.7. Traces of neuron sequences from actions in KTH dataset.

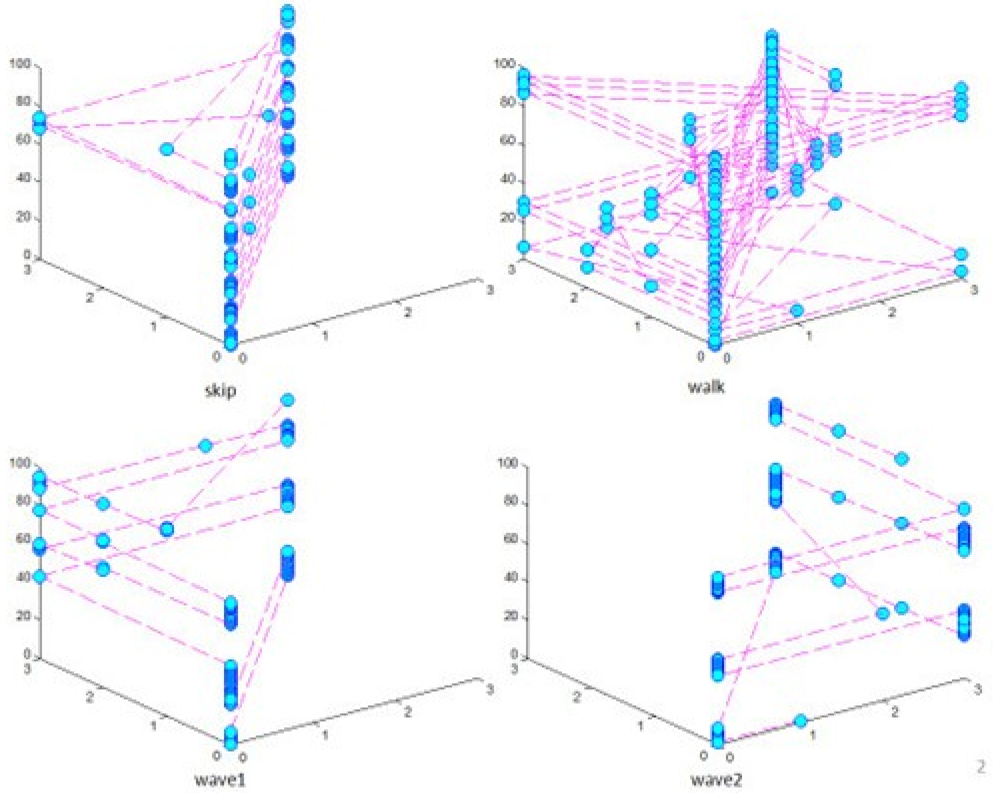


Figure 3.8. Traces of neuron sequences from actions in Weizmann dataset.

3.1.5. Dynamic Model for Action Recognition

Learning spatio-temporal data is a complex procedure to analyze the correlation and variability across space and time. There is some limit on the estimation of traditional state- space models, since the high dimensional parameters lead to complicated dependency structures. The proposed dynamic model can optimize the parameters and train the ensemble-learning model for classification.

We assume the input data $X_t = \{X(x_i;t)\}$, $i = 1,...,S$, where S is the number of spatial data attributes at the time t. The covariance matrix of the zero-mean Gaussian noise is Δ_t . Θ_t describes the state transition over the time t.

We collect the dynamic model parameters as $\Phi = \{\Theta_t, \Delta_t\}$. The primary goal of this model is to estimate the modeling parameters through expectation-maximization (EM). From Algorithm 1, we can obtain the discrete label sequences D^* . The likelihood of the input data sequences can be estimated as below:

$$P(D^* \mid \Phi) = \prod_{i=1}^{n} P(D_i^* \mid \Phi) \tag{3.11}$$

According to Eq. (3.11) the log likelihood can be expressed as

$$\log P(D^* \mid \Phi) = \sum_{i=1}^{n} \log P(D_i^* \mid \Phi) \tag{3.12}$$

Considering the hidden variable η in the latent space, the log likelihood can be recalculated in

$$\log P(D^* \mid \Phi) == \sum_{\eta} \sum_{i=1}^{n} \log P(D_i^*, \eta \mid \Phi) \tag{3.13}$$

Suppose ξ is the probability distribution of the hidden variable η, the target function $F(\xi, \Phi)$ in EM can be described as

$$F(\xi, \Phi) = E_{\xi}[\log P(D_i^*, \eta) \mid \Phi] - E_{\xi}[\log(\xi(\eta))] \tag{3.14}$$

The target function expression in Eq. (3.14) indicates the ``free energy" in statistical physics via the expected energy and the entropy of the distribution ξ. The main steps in the iterative procedure of EM are focused on maximizing the target function $F(\xi, \Phi)$ respectively. At time *t*, the E step fixes Φ and selects ξ to maximize $F(\xi, \Phi)$. Based on the selection of ξ in the E step, the M step selects Φ to maximize $F(\xi, \Phi)$.

E step: $\xi^{(t+1)} \leftarrow \arg\max_{\xi} F(\xi, \Phi^{(t)})$

M step: $\Phi^{(t+1)} \leftarrow \arg\max_{\Phi} F(\xi^{(t+1)}, \Phi)$

We can predict the class label based on below:

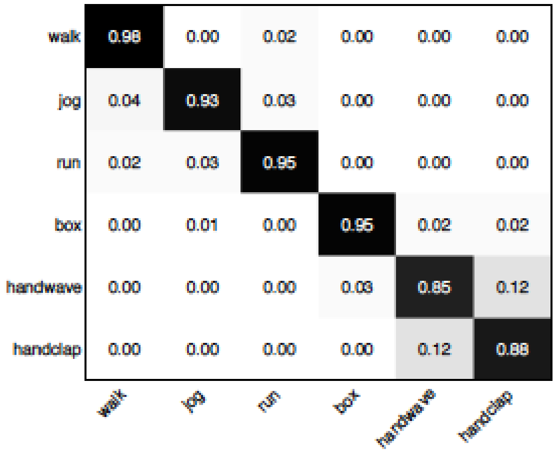
$$y = \arg\max_{s_i \in S} \sum_{j=1}^{n} P(D^* \mid \overline{\Phi_j}, s_j) P(s_j \mid \overline{\Phi_j}) P(\overline{\Phi_j}) \tag{3.15}$$

where $\overline{\Phi_j}$ represents one of the alternative models, *S* is the set of all class labels.

3.1.6. Experiments and Performance

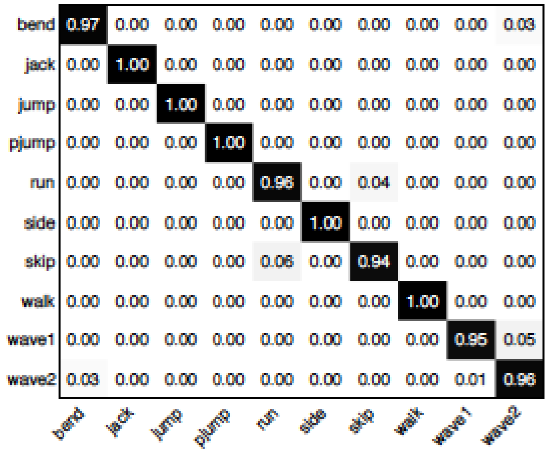
KTH [39], Weizmann action [45] and UCF sport datasets [46] are used to evaluate the performance of the proposed method. To analyze the effects of periodic and non-periodic actions, we calculate optical flow in feature extraction [47]. Optical flow approximates local image motion based on local derivatives in a video sequence, and it can essentially reflect the spatio-temporal variability between two consecutive frames.

The performance of the proposed approach can be analyzed through the confusion matrix. In KTH dataset, “walk” can be easily recognized with the rate of 98% in Figure 3.8., but it is confused by “run” with 2%. “jog” and “run” are both affected by “walk”. “handwave” and “handclap” affect the recognition results with each other. From Figure 3.9., we can see that our method achieves 100% accuracy for recognizing the actions including “jack”, “jump”, “pjump”, and “side”. There are some errors in other actions. For example, both “bend” and “wave2” are the actions with two-hand up in some specified scenarios. The spatial similarities over time make it difficult to achieve high accuracy. As shown in Figure 3.10, the sport action recognition is also a challenging task. We can recognize the action “dive” with high accuracy, but it becomes more difficult to recognize other actions, such as “run”. Although the spatio-temporal dynamic topological structure improves dynamic model to make accurate decision, the false recognition occurs when training frame snapshots or sequences shares the similar variations of spatio-temporal features.



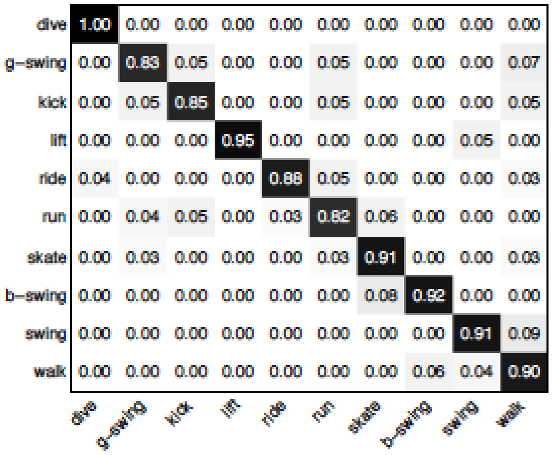
	walk	jog	run	box	handwave	handclap
walk	0.98	0.00	0.02	0.00	0.00	0.00
jog	0.04	0.93	0.03	0.00	0.00	0.00
run	0.02	0.03	0.95	0.00	0.00	0.00
box	0.00	0.01	0.00	0.95	0.02	0.02
handwave	0.00	0.00	0.00	0.03	0.85	0.12
handclap	0.00	0.00	0.00	0.00	0.12	0.88

Figure 3.8. Confusion matrix on KTH action dataset.



	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03
jack	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
jump	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pjump	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
run	0.00	0.00	0.00	0.00	0.96	0.00	0.04	0.00	0.00	0.00
side	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
skip	0.00	0.00	0.00	0.00	0.06	0.00	0.94	0.00	0.00	0.00
walk	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
wave1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.05
wave2	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.96

Figure 3.9. Confusion matrix on Weizmann action dataset.



	dive	g−swing	kick	lift	ride	run	skate	b−swing	swing	walk
dive	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
g−swing	0.00	0.83	0.05	0.00	0.00	0.05	0.00	0.00	0.00	0.07
kick	0.00	0.05	0.85	0.00	0.00	0.05	0.00	0.00	0.00	0.05
lift	0.00	0.00	0.00	0.95	0.00	0.00	0.00	0.00	0.05	0.00
ride	0.04	0.00	0.00	0.00	0.88	0.05	0.00	0.00	0.00	0.03
run	0.00	0.04	0.05	0.00	0.03	0.82	0.06	0.00	0.00	0.00
skate	0.00	0.03	0.00	0.00	0.00	0.03	0.91	0.00	0.00	0.03
b−swing	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.92	0.00	0.00
swing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.09
walk	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.04	0.90

Figure 1.10. Confusion matrix on UCF action dataset

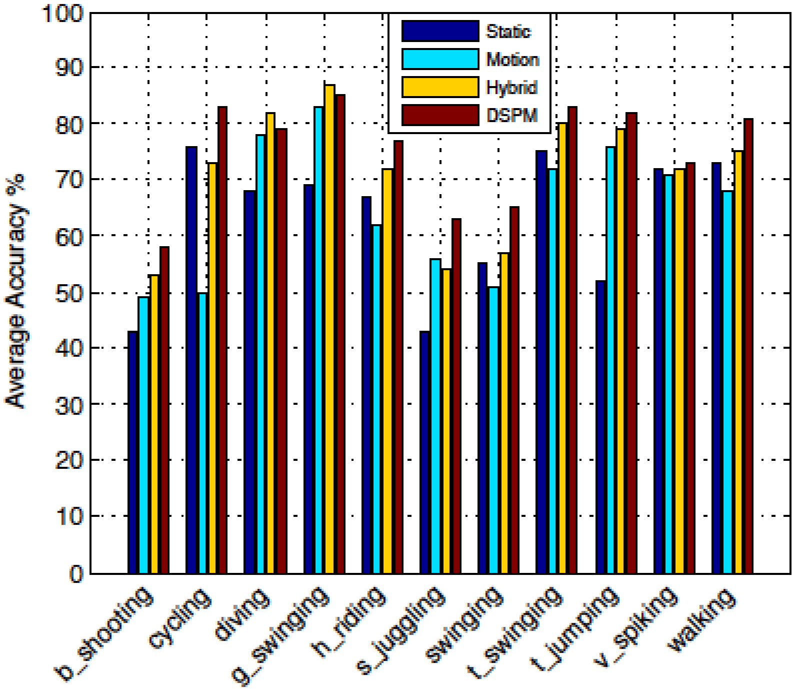
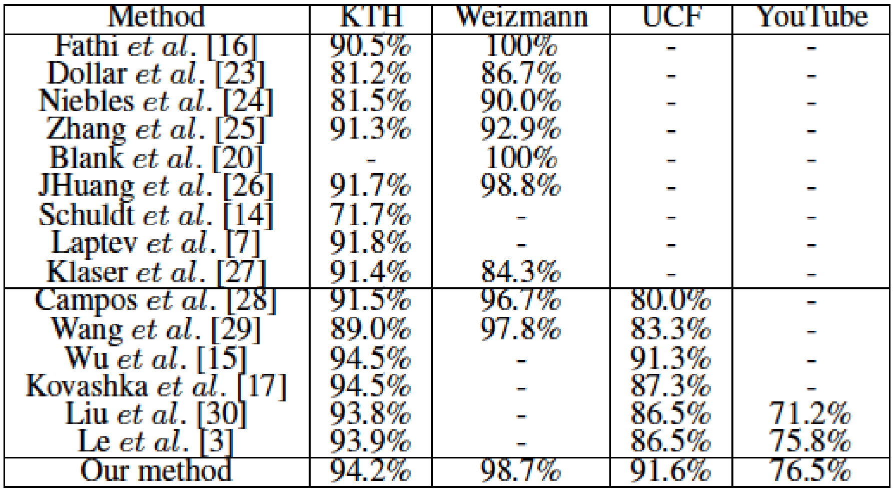


Figure 3.11. Recognition performance on YouTube dataset.

Table 3.1 Average accuracy on KTH, Weizmann, UCF, and YouTube datasets



Method	KTH	Weizmann	UCF	YouTube
Fathi *et al.* [16]	90.5%	100%	-	-
Dollar *et al.* [23]	81.2%	86.7%	-	-
Niebles *et al.* [24]	81.5%	90.0%	-	-
Zhang *et al.* [25]	91.3%	92.9%	-	-
Blank *et al.* [20]	-	100%	-	-
JHuang *et al.* [26]	91.7%	98.8%	-	-
Schuldt *et al.* [14]	71.7%	-	-	-
Laptev *et al.* [7]	91.8%	-	-	-
Klaser *et al.* [27]	91.4%	84.3%	-	-
Campos *et al.* [28]	91.5%	96.7%	80.0%	-
Wang *et al.* [29]	89.0%	97.8%	83.3%	-
Wu *et al.* [15]	94.5%	-	91.3%	-
Kovashka *et al.* [17]	94.5%	-	87.3%	-
Liu *et al.* [30]	93.8%	-	86.5%	71.2%
Le *et al.* [3]	93.9%	-	86.5%	75.8%
Our method	94.2%	98.7%	91.6%	76.5%

To verify the recognition capability of the proposal method, Table 3.1. shows the recognition results of many comparable approaches based on KTH, Weizmann and UCF dataset, respectively. On KTH dataset, Wu *et al*. [40] and Kovashka *et al*. [42] achieved the best performance with 94.5%. Our method can achieve 94.2% on average. On Weizmann dataset, Blank [45] and Fathi [41] achieved 100%, Jhuang [48] achieved 98.8%, and our method achieved 98.7%. On the most challenging UCF dataset, Kovashka *et al*. [42] and Wu *et al.* [40] achieved 87.3% and 91.3%, respectively. Our method with 91.6% performs better than these methods. The performance of our method is comparable with these state of the art methods on action datasets. Particularly, for more complex dataset, such as UCF sport dataset, our method can effectively improve the recognition performance. But more importantly our method can adaptively learn from low level features, such as optical flow, rather than using strong features. This improves model robustness, and requires less human intervention.

To further analyze the robustness of our proposed method on challenging realistic actions, we compare our method with the work by Liu et al. based on UCF YouTube dataset with 11 action categories. This video dataset is very challenging, including mixture of steady and shaky cameras, diversity of background, different viewpoint, various illumination and low resolution. Figure 3.11. shows the recognition accuracies of three variations of method in [49] and our DSPM. From Figure 3.11. we can see that DSPM outperforms the average recognition accuracy in previous work, especially for some difficult scenarios such as “s_juggling”, “swinging” and “b_shooting”. The experiment results indicate that our method performs particularly well in recognizing cyclic actions such as “s juggling”, “swinging”, “cycling”, “t jumping” and “walking”. DSPM has the capability to handle this problem if simple redundancy detection is employed. The similar performance on “v_spiking” indicates the effectiveness of DSPM to recognize group action of multiple people. Through the comparison in Table 3.1., DSPM can perform as a competitive method compared with the existing methods.

3.2. Small Human Group Activity Modeling Based on Gaussian Process Dynamic Model and Social Network Analysis (SN-GPDM)

3.2.1. Introduction

Human action recognition has been studied for decades in the computer vision filed, as it can be applied to many surveillance systems. Most current human action recognition research works focus on single human action identification. Many researchers [50][51][52] have tested their algorithms on two popular data sets: Weizmann human action dataset [45] and KTH human action dataset [39], and the experiment results showed that most algorithms could achieve a very high recognition rate. As the importance for public safety increases, much more attention are needed for recognizing interactions between people. Thus group activity recognition has become an essential issue in action recognition. However, most current group activities recognition works are concentrated on dense human crowd analysis. Mehran et.al [53] proposed a social force model for abnormal crowd behavior detection. Wang et.al [54] also proposed an unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models. All aforementioned methods computed different motion features over the entire frame for activity recognition.

Besides single human action recognition and human crowd analysis, small human group (around ten people) action recognition, has more practical applications in developing realistic surveillance systems. As shown in Figure 3.12 most public safety scenarios consist of small group activities. However, relatively few research has been done on this topic, due to the difficulties of describing varying number of participants and the mutual occlusion between people. Recently, Ni et.al [55] introduced three types of localized causalities for human group activities with different number of people, and their experiment results showed that intro-person feature could be used to classify group actions. They provided feature vectors of different sizes to describe different group activities, which needed to train specific classifiers using different input samples with different lengths. Chang et.al [56] proposed a bottom-up method to form a group and calculated the similarity of different groups. Ge et.al [57] also developed a hierarchical clustering algorithm for small group detection in a crowded scene. Guimera et.al [58] proposed a collaboration network structure to determine the team performance, and the experiment result indicated that team assembly mechanism could be used for predicting and describing the group dynamics.

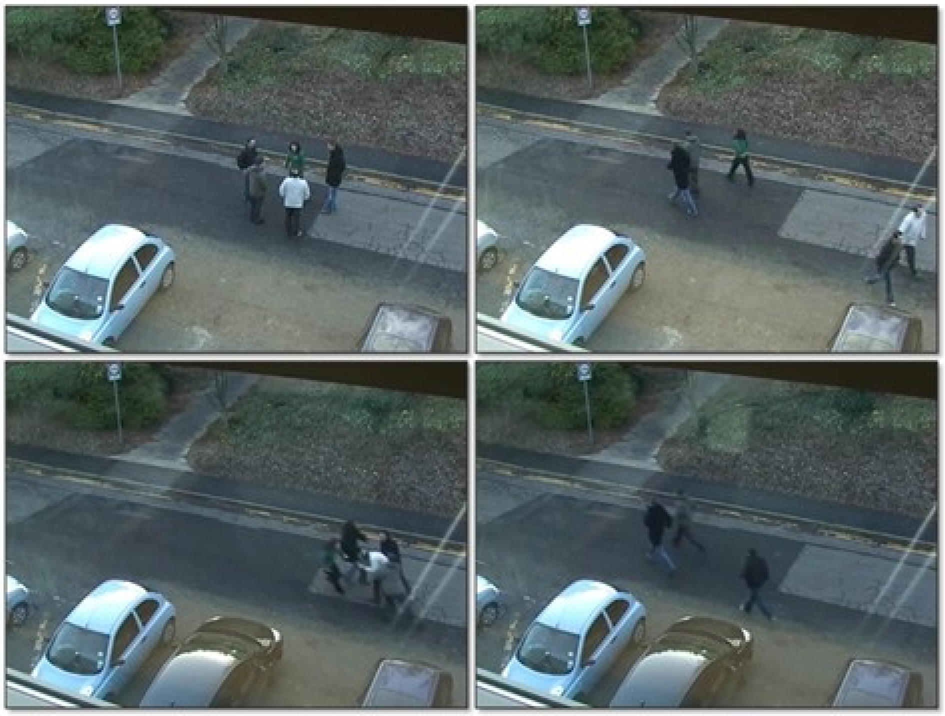


Figure 3.12. BEHAVE dataset samples (In_Group, Group_Split, Group_Fight, Chasing)

In contrast with single person action recognition, small groups contain much richer inter-person interactions among group members. Compared to crowd analysis, in which each person can be regarded as a point in a flow, small groups contain much detail information about each individual in the group. Major challenges of small group activity analysis include mutual occlusions between different people, the varying group size, and the interaction within or between groups. Therefore small group activity recognition demands a structural feature to bridge the local description of single human and global description for crowd analysis, as well as addressing both the spatial dynamics (varying group size) and temporal dynamics (varying clip length). Unlike single person or dense crowd analysis, small group action recognition require detection and tracking of each group member rather extracting feature from the entire scene, as there may be several small groups with different actions in an individual scene.

Extending recent works [59][60][61] on single human motion modeling by Gaussian Process Dynamic Models, we propose a novel structural feature set to represent group activities as well as a probabilistic framework for small group activity learning and recognition. Our framework consists of four stages, as shown in Figure 3.13.

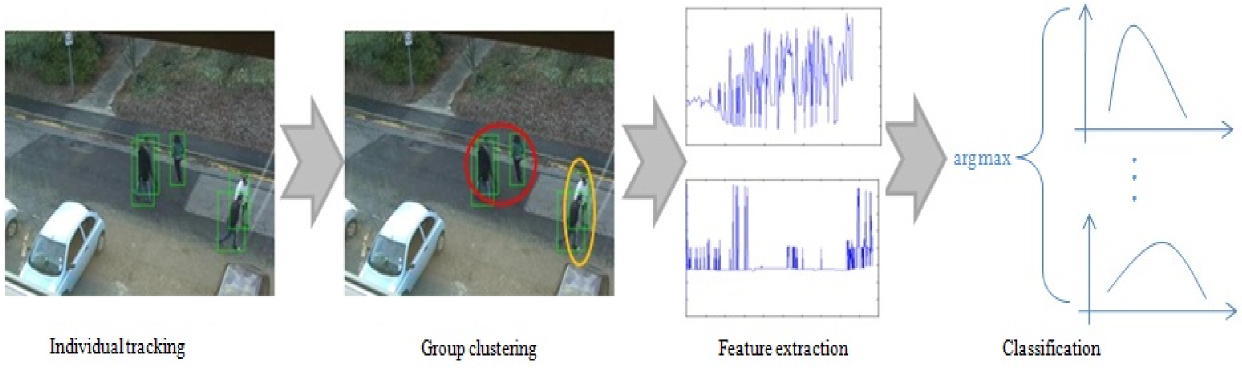


Figure 3.13. Human group action recognition framework.

First, we apply a robust mean-shift [62] based tracker to track each individual in a small group sequentially. Second, the output coordinates of each tracker will be clustered and allocated to different small groups. Based on social network feature description, we extracted the structural features from each video clip in the third stage from each video clip. Those feature vectors contain global structure of each group as well as local motion description of each group member, and they all have same size regardless the different number of people inside each group.

In the last stage, the feature vectors from each frame will form a feature matrix for each video clip. A Gaussian process dynamical model is trained to model different group behaviors respectively. The group activity matrix will be projected to a low dimensional latent space and get a compact representation. A posterior conditional probability is compute with each trained model to identify different group behaviors. We validate our framework on two publicly available data sets: BEHAVE data set [25] and IDIAP data set [63].

Our main contributions are listed as follows: First of all, we proposed a social network analysis based structural feature set to represent the dynamic of small group people. The structural feature characterizes both the global distribution of a group as well as local motion of each individual. In addition, this feature set can keep a fixed length while handling vary group size and group location, which is very important for recognition. Secondly, we established a probabilistic framework for human behavior classification, which extended the GPDM [60] to address classification. Different specific GPDM is pre-trained for each group activity, then the conditional probability is computed for the new coming activity feature, and the one with the highest probability is selected as the group activity type. As there is no length constraint for input training and testing feature, this GPDM based recognition framework can address recognition of video clips with different length. Therefore, our proposed model can represent the dynamical characteristic of the similar activities with different time duration. The difference between our proposed model with GPDM in [59][60] is our model introduce the conditional property of GPDM for classification, while GPDM in [59][60] are mostly used for single human motion reconstruction.

3.2.2. Social Network Analysis Based Feature Set

Feature extraction plays an essential role in human action recognition. Most features used for human action recognition fall into two big categories: general low level feature and middle level feature. General low level feature includes human motion, optical flow, 3D SIFT [64] (Scale Invariant Feature Transform) or STIP [32] (Spatial Temporal Interest Points), which are directly computed on the entire image region. General low level features are good for single person action classification. As small group human behavior involving interactions between different members, it needs features capture local detail information as well as global structure description. Thus middle level feature [55], which characterizes the group structure information above general low level feature, has been developed for small group human action recognition.

Social network analysis [65] was originally designed to model the social structure of individuals and relationships among people in real world societies. It maps the social individuals or "actors" as nodes and relationships between them as links to form a graphic based network. Inspired by the social network analysis, we proposed a set of structure features to capture the dynamic properties of a small group behavior. To our best knowledge, this is the first time that social network analysis is used to model group behavior in the surveillance videos.

Similar to the original definitions of *Betweenness*, *Closeness*, and *Centrality* [66][67] in social network analysis, we define several group structure features for human group activity recognition.

Group center:

Suppose there are n people in a group, the group center $m = \left(\frac{1}{n} \sum_{i=0}^{n} x_i, \frac{1}{n} \sum_{i=0}^{n} y_i \right)$ is defined as the mass center of the group.

Motion histogram:

Motion vector is defined as the position difference of each individual between two consecutive frames. For each person in a group, we can calculate the orientation and magnitude of the motion vector. Suppose there are n people in a group, then we have $\mathbf{M}_t = \{m_i\}_t (i = 1, \ldots, n)$, then the magnitude of m_i is accumulated into orientation histograms and normalized at each direction, as shown in the Figure 3.14. The length of each arrow is corresponding to the sum of the vector magnitude near that direction. As the orientation has been divided to 8 bins, the motion histogram is an 8-dimension vector for each group in each frame.

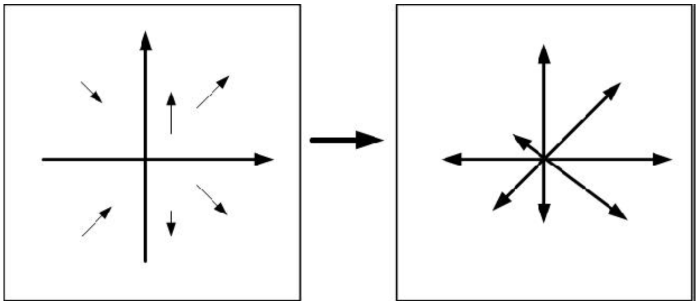


Figure 3.14. Example of motion histogram.

Closeness histogram:

Closeness describes how close an individual is near to all the other nodes, directly or indirectly in a network. In our experiment, closeness vector is defined as the directional vector between every two different people. Suppose there are n people in a group, then we have $\mathbf{C}_t = \{c_i\}_t (i = 1, \ldots, n)$. Similar to motion histogram, the magnitude of c_i is accumulated into 8-bin orientation histograms and normalized at each direction. The length of each arrow is corresponding to the sum of the vector magnitude near that direction. As the orientation has been divided to 8 bins, the motion histogram is an 8-dimension vector for each group in each frame.

Centrality histogram:

Centrality was originally used for describing the overall network structure based on each node's location in a network. In this paper centrality vector is defined as the directional vector which from the position of each person toward the group mass center. Suppose there are n people in a group, then we have $\mathbf{Ce}_t = \{ce_i\}_t (i = 1, \ldots, n)$. Similar as motion histogram, the magnitude of ce_i is accumulated into 8-bin orientation histograms and normalized at each direction. The length of each arrow is corresponding to the sum of the vector magnitude near that direction. As the orientation has been divided to 8 bins, the centrality histogram is a 8-dimension vector for each group in each frame.

Relative velocity histogram:

The relative velocity is defined as the velocity difference between each individual and the group center. Suppose there are n people in a group, group center's velocity is the group center difference between two consecutive frames. Each person's relative velocity $\mathbf{v}_t = \{v_i\}_t (i = 1, \ldots, n)$ is calculated as the person's velocity minus the group velocity. Relative velocity describe the group movement regardless the group size and group location in a scene, therefore relative group information can present the group movement. To better represent the relative motion distribution, the magnitude of v_i is accumulated into 8-bin orientation histograms and normalized at each direction. The length of each arrow is corresponding to the sum of the vector magnitude near that direction. As the orientation has been divided to 8 bins, the centrality histogram is an 8-dimension vector for each group in each frame.

As described above, for each frame, a 26 dimensional vector is extracted, including group center, motion histogram, closeness histogram and centrality histogram. Suppose the length of a group activity (total frame number) is m, then the size of the feature matrix is $26 \times m$.

3.2.3. Gaussian Process Dynamical Model and Conditional GPDM

According to previous section, suppose we have a group activity clip of m frames, and the size of the feature matrix is $26 \times m$. Figure 3.15 shows the centrality feature of two different group activities. The assumption of this paper is that in the normal situation, the motion distribution of a group is prone to have a Gaussian distribution. If we treat centrality feature in the Figure 3.15 as a Gaussian process, then the centrality histogram at each frame is a sampling of this process. Different group activities can be seen as a set of Gaussian processes with different means and covariance matrices. Therefore Gaussian process can be used to model the dynamics in the temporal dimension. However, the size of covariance matrix will increase as the number of samples increases. In addition, Gaussian process just captures the general properties of our proposed structural feature, more specific characteristic of human motion needs to be addressed.

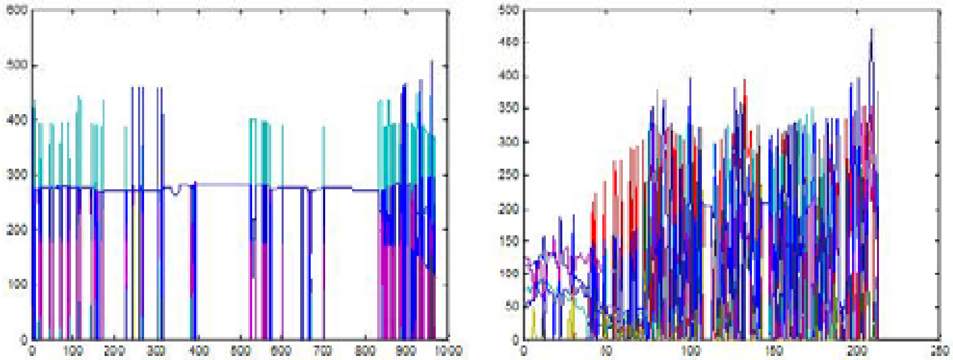


Figure 3.15. The overlapped central histogram of group talking (left) and group fighting (right) from two video clips

In order to describe the dynamic property of the group behavior, here we adopt Gaussian Process Dynamical Model(GPDM) to represent different group activities.

Gaussian Process Dynamical Model was derived from Gaussian Process Latent Variable Model (GPLVM) [68], which provided a probabilistic mapping from high-dimensional observation data to low-dimensional latent space and represented the joint distribution of observation data. Compared with other dimension reduction algorithms, such as LLE [69] and ISOMAP [70], GPLVM has the advantage to provide the posterior probability of the projected observation space. To address sequential data series in GPLVM, J. Wang et.al [60] introduced GPDM, which augmented the GPLVM by adding first-order Markov dynamic in the latent space. Consider a basic discrete model with first order Markov dynamics in equations below:

$$x_t \ = f(x_{t-1}, U) + \eta_{x,t} \tag{3.16}$$

$$z_t \ = h(x_t, V) + \varsigma_{z,t} \tag{3.17}$$

where x_t is the latent variable and z_t is the observation variable at time t. $\eta_{x,t}$ and $\varsigma_{z,t}$ are zero-mean, isotropic, white Gaussian distributed noise for the latent and observation spaces respectively.

Eq. (3.16) is the dynamic model in the latent space. Similar to GPLVM, the dynamic model also marginalizes the transform function U to predict X_{out}. With a first-order Markov dynamics, the joint probability density over the latent coordinates X_t is expressed as in Eq. (3.18).

$$p(X \mid \overline{\alpha}) = \frac{p(x_1)}{\sqrt{(2\pi)^{\frac{D(N-1)}{2}} \mid K_X \mid^{\frac{N}{2}}}} exp\left(-\frac{1}{2}(K_X^{-1} X_{out} X_{out}^T) \right), \tag{3.18}$$

where $X_{out} = [x_2, ..., x_N]^T$ are considered as testing data, and $\overline{\alpha}$ is a vector of kernel parameters. We assume $p(x_1)$ also has a Gaussian prior. K_X is the $(N-1) \times (N-1)$ kernel matrix constructed from $[x_1, ..., x_{N-1}]$, and a linear kernel in Eq. (3.19) is used.

$$k_X(x, x') = \alpha_1 x^T x' + \alpha_2^{-1} \delta_{x,x'} \tag{3.19}$$

Eq. (3.16) represents a non-linear projection from the latent space X to the observation space Z.

$$p(Z_t \mid X_t, \beta, \Omega) = \frac{\mid \Omega \mid^N}{\sqrt{(2\pi^{ND} \mid K_Z \mid^D)}} exp(-\frac{1}{2} tr(K_Z^{-1} Z \Omega^2 Z^T)) \tag{3.20}$$

where Ω is a scale parameter, N is the length of observation sequences Z, D is the data dimension of Z, K_Z is the kernel function.

In our study, RBF kernel is given by the following Eq. (3.21) and employed for the mapping between the observation and the latent space,

$$k_Z(x, x') = exp(-\frac{\gamma}{2} \| x - x' \|^2) + \beta^{-1} \delta_{x,x'} \tag{3.21}$$

where x and x' are any pair of variables in the latent space, γ controls the width of the kernel, β^{-1} is the variance of the noise.

According to all the equations above, we can derive the GPDM model $\{\Lambda\}$ as in the equation (7) based on the Gaussian priors, first order Markov dynamics and latent space mapping.

$$\Lambda \ = p(X,Z,\bar{\alpha},\bar{\beta},\Omega) \ = p(Z_t \mid X_t,\bar{\beta},\Omega)p(X \mid \bar{\alpha})p(\bar{\alpha})p(\bar{\beta})p(\Omega) \tag{3.22}$$

Given a trained GPDM, $\Lambda = \{Z^T, X^T, \bar{\alpha}, \bar{\beta}, \Omega\}$, where Z^T is the training observation data, X^T is the corresponding latent variable sets, $\bar{\alpha}$ and $\bar{\beta}$ are hyperparameters vectors, and Ω is a scale parameter. The conditional probability of a new observation $Z^{\star}$ can be defined in Eq. (3.23).

$$\begin{aligned} p(Z^{(*)}, X^{(*)} \mid \Lambda) &= p(Z^{(*)} \mid X^{(*)}, \Lambda)p(X^{(*)} \mid \Lambda) \\ &= \frac{p(Z, Z^{(*)} \mid X, X^{(*)}, \bar{\beta}, \Omega)}{p(Z \mid X, \bar{\beta}, \Omega)} \frac{p(X, X^{(*)} \mid \bar{\alpha})}{p(X \mid \bar{\alpha})} \\ &\propto p(Z, Z^{(*)} \mid X, X^{(*)}, \bar{\beta}, \Omega)p(X, X^{(*)} \mid \bar{\alpha}) \end{aligned} \tag{3.23}$$

Suppose the length of Z and $Z^{(*)}$ is N and M, then the kernel size of $\{Z, Z^{(*)}\}$ is $(N+M)\times(N+M)$. To reduce the computational cost, we define two kernel matrices: $Q_{i,j} = k_Z(x, x^{(*)})$, and $R_{i,j} = k_Z(x^{(*)}, x^{(*)})$, then we can derive $p(Z^{(*)} \mid X^{(*)}, \Lambda)$ as:

$$\begin{aligned} & p(Z^{(*)}, X^{(*)} \mid \Lambda) \\ &= \frac{|\Omega|^M}{\sqrt{(2\pi)^{MD} |K_{Z^{(*)}}|^D}} exp\left(-\frac{1}{2} tr(K_{Z^{(*)}}^{-1} P_Z \Omega \Omega^2 P_Z^T)\right) \end{aligned} \tag{3.24}$$

where $P_Z = Z^{(*)} - Q^T K_Z^{-1} Z$ and $K_{Z^{(*)}} = R - Q^T K_Z^{-1} Q$.

Similar to the derivation above, we define two other kernel matrices: $O_{i,j} = k_X(x, x^{(*)})$, and $L_{i,j} = k_X(x^{(*)}, x^{(*))}$,then we can obtain $p(X^{(*)} \mid \Lambda)$ as:

$$p(X^{(*)} \mid \Lambda) = \frac{p(x_1^{(*)})}{\sqrt{(2\pi)^{(M-1)d} |K_{X^{(*)}}|^d}} exp\left(-\frac{1}{2} tr(K_{X^{(*)}}^{-1} H_X H_X^T)\right) \tag{3.25}$$

where $H_X = X_{out}^{(*)} - O^T K_X^{-1} X_{out}$ and $K_{X^{(*)}} = H - O^T K_Z^{-1} O$.

During the updating process, $K_{Z^{(*)}}$ and $K_{X^{(*)}}$ only need to be inverted once. It should be noted that, the length of new observation can be different with the size of training data.

3.2.4. Proposed Framework for Behavior Classification

As shown in Figure 3.13, our small group activity recognition framework consists of four stages: adaptive mean-shift tracking, small group clustering, group feature extraction and group activities recognition.

Adaptive Mean-shift Tracking

One of the important factor for small group human activities analysis is the accuracy and robustness of tracking each individual in the group. As the development of multiple camera systems, the accurate tracking of each individual can be well addressed. In this paper we apply adaptive mean-shift tracking on the two data sets.

Compared to general mean-shift tracking, on-line feature selection is applied during the adaptive mean-shift tracking. In [62], the feature consisted of linear combination of pixel valves at R,G,B channels: $F \equiv \omega_1 R + \omega_2 G + \omega_3 B$, where $\omega_i \in [-2,-1,0,1,2], i = 1,\dots,3$. By pruning all redundant coefficients of ω_i, the feature set was cut down to 49. Linear discriminative analysis (LDA) was then used to determine the most descriptive feature for target tracking.

In order to reduce the computational complexity during tracking, we just update the feature set every 50 frames instead of updating the feature set at each frame. In addition, we extend the single mean-shift tracking algorithm for multiple targets tracking. As the cameras were fixed in these two data sets, a simple motion detector is applied to detect each new person coming into scene. Once a person comes in the scene, a new tracker will be allocated and track that person overtime. Since our focus of this paper is not reliable multiple targets tracking, we just reinitialize each target manually if the tracking algorithm fails for some reason.

Small Group Clustering

After obtaining all the positions of each target, a group clustering algorithm [56] will be applied to locate small groups. We first calculate the closeness of each person and use the Minimum Span Tree (MST) clustering to obtain the distribution of each group.

After that, we follow the hierarchical clustering method described in [56] to locate the mass center of each small group.

Small Group Activity Recognition

The small group activity recognition can be divided to two phases: group activity training and group activity classification. In the training stage, for each small group activity $\{ A_i, i = 1,\dots,n \}$, a GPDM $\{ \Lambda_i, i = 1,\dots,n \}$ will be trained. Suppose we have k samples of a group activity A_i, the length of each sample is m, then we have k feature matrices of size $26 \times m$. To learn a specific GPDM for A_i, we will first compute the mean value $\overline{Z}$ of k feature matrices, and utilize the mean for training.

GPDM is applied to learn the specific trajectories of a group activity. The probability density function of latent variable X and the observation variable $\overline{Z}$ are defined by the following equations. The basic procedure Gaussian Process Dynamical Model training is described as below:

1. Creating GPDM:
   GPDM $\Lambda = \{\overline{Z}^T, X^T, \overline{\alpha}, \overline{\beta}, \Omega\}$ is created on the basis of the trajectory training data sets, i.e. extracted structural feature, where $\overline{Z}^T$ is the training observation data, X^T is the corresponding latent variable sets, $\overline{\alpha}$ and $\overline{\beta}$ are hyperparameters.

2. Jointly initializing the model parameters:
   The latent variable sets and parameters $\{X^T, \overline{\alpha}, \overline{\beta}\}$ are obtained by minimizing the negative log-posterior function $-lnp(X^T, \overline{\alpha}, \overline{\beta}, \Omega \mid \overline{Z}^T)$ of the unknown parameters $\{X^T, \overline{\alpha}, \overline{\beta}, \Omega\}$ with scaled conjugate gradient (SCG) on the training datasets.

3. Train GPDM for each group activity:
   For each group activity $\{A_i, i = 1, \ldots, n\}$, repeat the procedure 1 and 2, create a corresponding GPDM: $\{\Lambda_i, i = 1, \ldots, n\}$.

After training, we have a set of GPDMs: $\{\Lambda_i, i = 1, \ldots, n\}$ for the human group activities. When a new human group activity $Z^\star$ coming in, we will compute the conditional probability with respect to each trained GPDM, and select the one with highest conditional probability.

1. Calculate the conditional probability with each trained GPDM:
   For each trained GPDM $\{\Lambda_i\}$, compute $X_i^\star$ by using the learned parameters: $\{\overline{\alpha}_i, \overline{\beta}_i\}$. This can be obtained by minimizing the negative log-posterior function $-lnp(X^T, \overline{\alpha}_i, \overline{\beta}_i, \Omega \mid Z^\star)$ with scaled conjugate gradient (SCG) on the training datasets. After that, we can calculate the conditional probability $P\left(Z_i^{(*)}, X_i^{(*)} \mid \Lambda_i\right)$ by Eq. (3.24).

2. Select the GPDM with the highest conditional probability:
   The new group activity can be determined by the following equation:

$$\underset{i=1,\ldots,n}{argmax}\, P\left(Z_i^{(*)}, X_i^{(*)} \mid \Lambda_i\right) \tag{3.26}$$

As we discussed in the previous section, the length of new observation can be different with the size of training data, which means that the number of frames in test clips can be different with training clips. Therefore our trained model can address the dynamics in the temporal dimension. As the duration of an activity may change under different situation, it is important that the classifier can handle the testing sequences with varying lengths.

3.2.5. Experimental Results

We test our framework on two popular group activity data sets. The first one is the recently released BEHAVE data set [25], which contains the ground truth for each group activity. The second data set is IDIAP data set [63], which was originally captured for multiple human tracking.

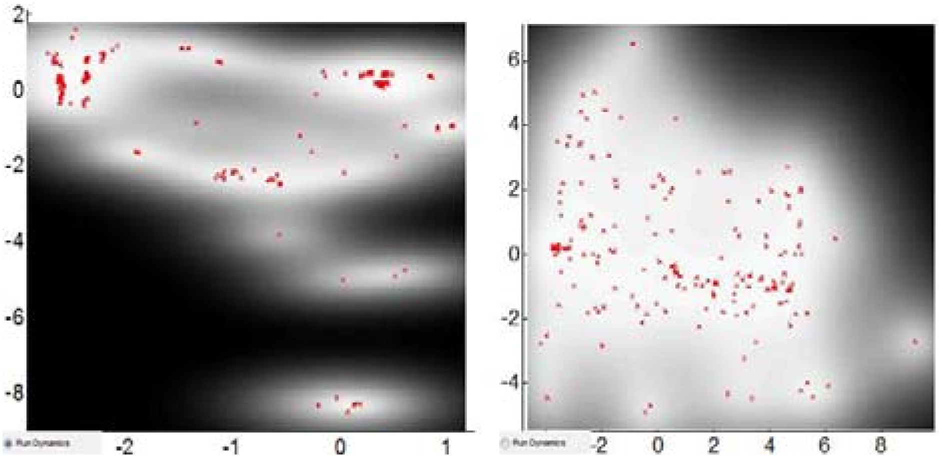


Figure 3.16. Visualization of trained GPDMs, the left one the InGroup, and the right one is Group Fight

Results on BEHAVE data set

The BEHAVE data set consists of four video clips, and 76, 800 frames in total. This video data set is recorded at 26 frames per second and has a resolution of 640 × 480. Different activities include: InGroup, Approach, WalkTogether, Split, Ignore, Following, Chase, Fight, RunTogether, and Meet. Examples are shown in Figure 3.17.There are 174 samples of different group activities in this dataset. As our focus is the small group activity analysis, we select 118 samples from all the group activities data set, and all the samples contain three or more people in the scene. The selected group activities include InGroup (IG), WalkingTogether (WT), Split (S) and Fight (F) as our group activities. For each activity, we divide the samples to ten-fold; with nine-fold for training and one fold for testing, the classification result is shown in the Table 3.2. Two of learned GPDMs are shown in the Figure 3.16. Each point in the latent space indicates one of the feature vectors in a frame. The distribution of InGroup activity is prone to have some local clusters in the latent space, while the distribution of GroupFight activity is similar to a random distribution.

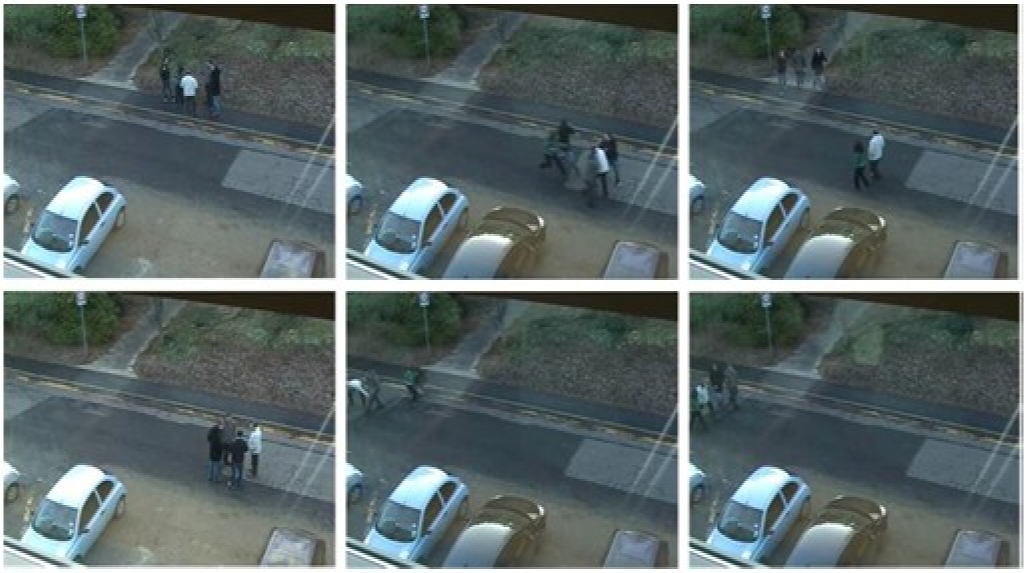


Figure 3.17. Sampling frames of InGroup and GroupFight, the left column the InGroup, the middle column is GroupFight, and the right column is WalkTogether

Table 3.2. Classification results of our method

	IG	WT	F	S
Our method	94.3%	92.1%	95.1%	93.1%

We also compare our results with the classification results in [25]. As in [25], the training and testing data is divided to 50/50, our proposed method can achieve 93.1%, comparing to 92.1% of HMM based method [25]. It should be noted that, the recognition rate is the average rate for all the activities, and the window size for calculating feature in [25] is 60. In addition, our proposed algorithm can adaptively recognize human group action with different length, although the method in [25] can reach a higher recognition rate when window size is increased to 100.

Results on IDIAP data set

IDIAP data set is firstly used in [63] for multiple targets tracking. The data set contains 37182 frames in total. We manually select 46 clips with different lengths for human group activity recognition. As there is no Fight activity in the IDIAP data set, we just evaluate three group activities: InGroup, WalkTogether, and Split. To validate the robustness of our framework, we directly apply the trained GPDMs from the BEHAVE data set for activity recognition on the IDIAP data set, and the overall average classification rate is 92.3%. The excremental results indicate that our proposed framework is robust to identify human group activities under different scenarios.

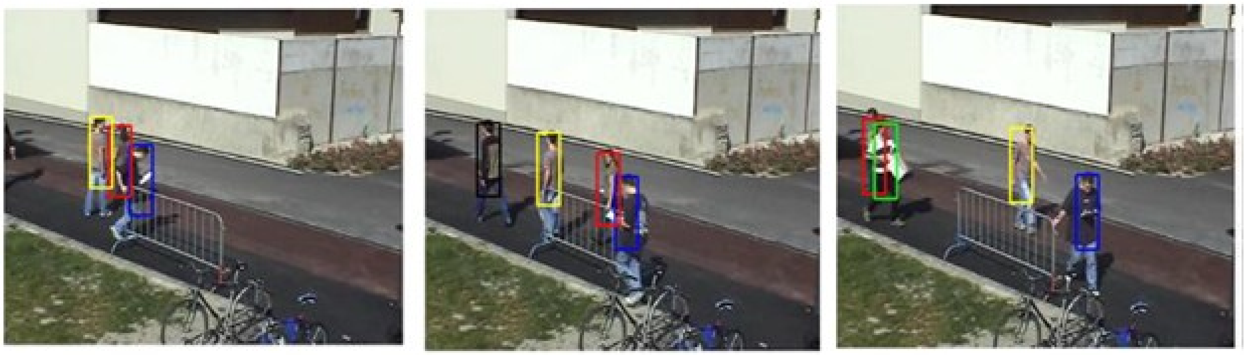
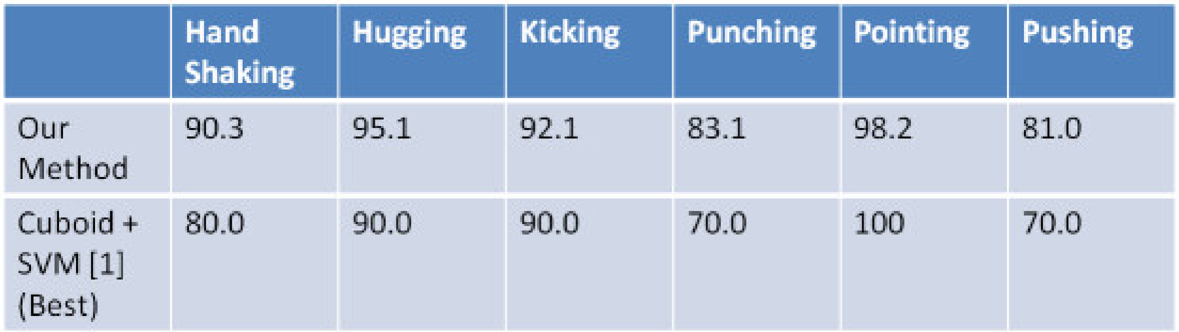
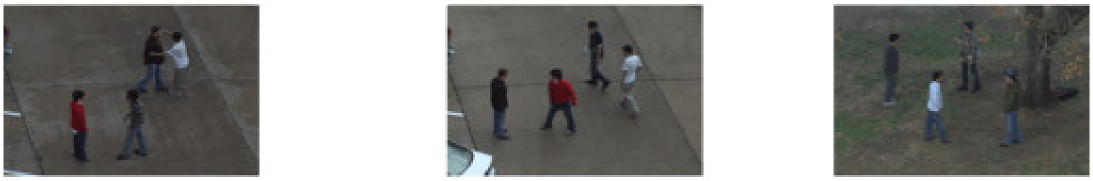


Figure 3.18. Sampling frames of InGroup, WalkTogether and Split frames from IDIAP data set are shown in the Figure 8.

We also tested our method on a new dataset, i.e. ICPR 2010 High-level Human Interaction Recognition Challenge dataset. Our results in Figure 3.19 are very competitive.



	Hand Shaking	Hugging	Kicking	Punching	Pointing	Pushing
Our Method	90.3	95.1	92.1	83.1	98.2	81.0
Cuboid + SVM [1] (Best)	80.0	90.0	90.0	70.0	100	70.0



[1] ICPR2010 Contest on Semantic Description of Human Activities : High-level Human Interaction Recognition Challenge

Figure 3.19. Experimental results on ICPR 2000 dataset.

3.3. Human Object Interactions Using SN-GPDM

3.3.1. Introduction

Many visual semantic concepts involve human and object interactions. Such scenarios can not be easily described by separated modeling of human and objects. We propose a new approach that extends our social network analysis based human group modeling method to capture human object interaction.

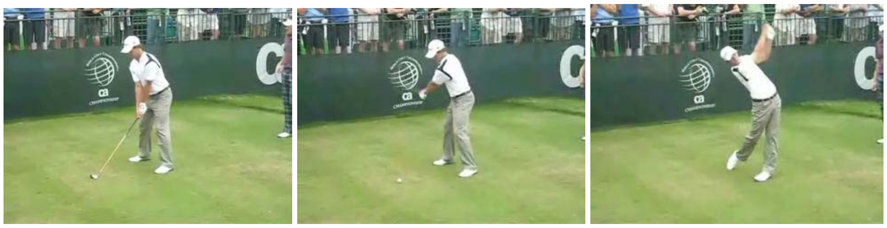
Human action understanding is a challenge topic and has been widely studied in applications such as surveillance and video retrieval. Many methods [52][71][72] have achieved high performance on recognizing single human with periodical actions in clear background scenarios, such as Weizmann human action dataset and KTH human action dataset. With increasing demands on video content analysis, studies have been more focused on complicated scenarios. A recent work by Yin et al. [73] studied the interactions among people based on BEHAVE dataset, which is a recorded data set with interactions within or between small groups, such as fighting, chasing, walking together and etc. These sequences are very close to real surveillance video. However there are more challenges lying in realistic videos, mostly sports and movie clips, which involve the interactions between human and objects.

In the study of recognizing human-object interactions, many researchers started from still images [74][75][76]. These existing methods on learning the interactions from static images are mostly using contextual information to build the relations between the object and human poses. Desia et al [74] provided a unified model based on detecting spatial contextual relations of multiple objects. Yao and Fei-Fei [75] presented a mutual context model to jointly model the human poses with objects in still images by two contextual information, which are the co-occurrence statistics and the spatial context between objects and body part. And Prest et al. [76] introduced a weakly supervised algorithm to learn the object relevant for the action and its spatial relation to the human.

Some recent attempts have been made on recognize interactions between human and object in videos. Gupta et al. [77] added the psychological analyses of human perception to a Bayesian model to recognize objects and actions in videos in a fully supervised manner. Prest et al. [78] further developed their method on realistic videos based on [76], by including spatio-temporal annotations about object's locations and human actions. Another work by Si et al. [79] provided an AND/OR grammar based algorithm to semantically understand certain human daily activities in office.

There are many challenges lying in the task of precisely identify the interactions between human and object in realistic videos. First, most existing methods require robust detection or tracking on human and objects, since the inconsistent information on human/body parts causes poor estimations on human poses and object positions. However, these tasks are very difficult in realistic videos. For one thing, it is common to see self-occlusions of the human body parts, or occlusions of objects by human or other less relevant background like branches of the trees. Another potential concern is that the quality of the video may vary significantly. The moving trajectories of objects may temporarily be lost because of the relatively poor quality of the video.

The other reason of losing the trajectory of the object or human parts is when those parts reaching out of the camera field of view during the activity. These natural difficulties are illustrated in Figure 3.20. Second, different from surveillance video which has a fixed camera scene, camera motions in realistic video must be taken into account as it affects the human/objects locations and the motion trajectory patterns.



(a) Frame 1 (b) Frame 23 (c) Frame 48

Figure 3.20. Challenges on object detections in realistic videos. In a sequence of human playing golf, (a) two hands are over- lapped all the time. In (b), the golf club is invisible temporarily because of the fast motion speed and relatively poor quality of the video. In (c), the club is out of the scene. Besides, the camera itself is not fixed and the background is not still.

In this section, we introduce a novel framework of recognizing human-object interactions by considering the body parts and objects as nodes of social network graphs in the spatial dimension, and analyzing the features of the social network overtime to understand the video sequences. This framework consists of three stages. First is tracking the body parts and object, which provides the spatial information by a tracking algorithm of [80]. Second stage is constructing the social network graphs and extracting the SNA features to describe the temporal dynamic of an interaction in each sequence. This is inspired by Yin et al. [73], in which individual humans were modeled as nodes in social networks and hence the SNA feature set were used to describe small human group activities. At the last stage, two classifiers are applied to the feature vectors, namely, a K-means cluster followed by SVM and a Hidden Markov model classification. Each method reduces the length of feature vectors to a lower dimension. Experiments were conducted on typical sports activities from HMDB dataset [81].

The contribution of our work is threefold. First, this social network based framework characterizes the distribution of the activity globally as well as the distribution of each node in the social network. Second, the social network analysis based feature set dynamically organizes the body parts and object as nodes in a graph. It is able to handle various numbers of nodes as well as length of the sequence. Last but not least, this framework is able to tolerant missing information during the sequence. Therefore, by using the social network structured feature sets, it does not required strictly precise detections in the earlier stage, which is a major difficulty in realistic videos and many other scenarios.

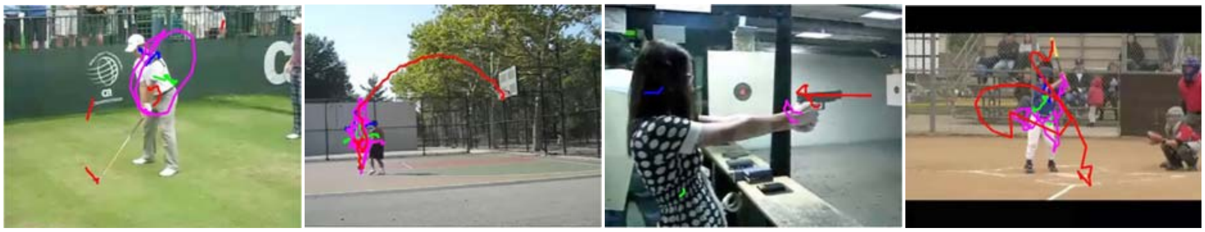
3.3.2. Human and Object Detection and Tracking

In our approach, the human object interaction is considered as a serial activities happening among the key body parts and the object, which we consider as nodes in a social network graph. It is a challenging task to have perfect detectors or trackers to obtain the precise locations of specific body parts and objects under realistic image quality conditions. In this framework, a reliable tracking algorithm is applies to obtain the locations for these node. We adopt a state-of-the-art tracking algorithm in [80] to have the motion trajectories. In human object interactions, we consider only a few crucial parts providing meaningful information and forming the social network as nodes. The body parts include head and upper-body centers, which represent the human positions in the frame, and hand positions, which are important to reveal the physical contact between human and object.

- **Human**: Detecting human body parts in realistic videos is particularly hard, because of self-occlusions as well as the variety of appearance and viewpoint. The method in [82] provides detection of 14 major joints on human body. We take a simplified body model and keep only a few crucial parts. The body parts considered as meaningful for understanding interactions include head and upper-body centers, which represent the human positions in the frame, and hand positions, which are important to reveal the physical contact between human and object.

- **Object**: Object detections are also very challenging. We use the detection approach presented in [83]. Training data are collected from Internet, and for each object class, there are around 100 clear images as positive data and another 100 images that randomly selected from Caltech-101 [84] as negative data.

Figure 3.21 shows some examples of the activity trajectories. The trajectories of head, upper-body center and hands are colored in blue, green and magenta respectively. The red color represents the object motion path. It may discontinue in some places due to the occlusions or the limitation of the video data. However, the proposed social network analysis based framework is robust enough to handle such missing information.



(a) golf (b) shoot ball (c) shoot gun (d) swing baseball

Figure 3.21. Examples of activity trajectories of the body parts and the objects. Blue and green lines are the trajectories of the head and upper-body center. Magenta represents hands trajectory and red color is for the object.

3.3.3. Social Network Analysis based HOI feature

For the recognition of human object interactions, we need structural information of the activities which can represent the interactions between body and object in a higher level. Inspired by the theoretic analysis of the social network [85] and its extensions on group activity recognitions [73], we introduce a new set of features to describe the dynamic properties of the human object interactions. Figure 3.22 shows the overview of this approach. To our best knowledge, this is the first time of using social network analysis based features to model human-object interactions.

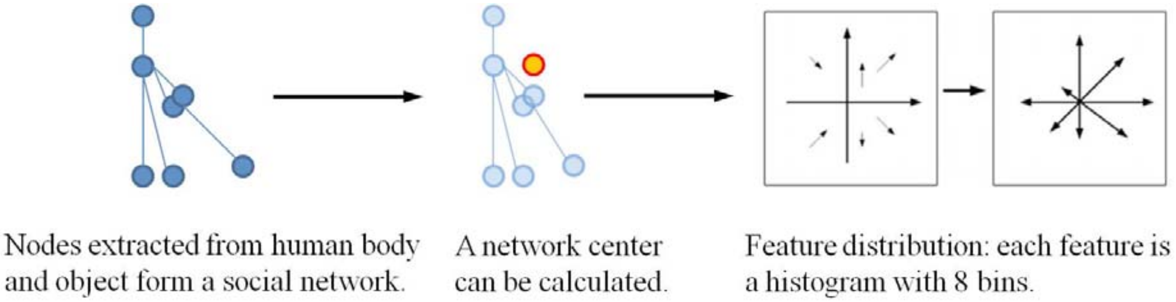

Nodes extracted from human body and object form a social network.
A network center can be calculated.
Feature distribution: each feature is a histogram with 8 bins.

Figure 3.22. Example of SNA based features on human-object interactions. The social network center is first calculated and other features are histograms distributed in 8 bins.

Network center:

Suppose there are n nodes in a network, the center $m_c = \left(\frac{1}{n}\sum_{i=0}^{n} x_i, \frac{1}{n}\sum_{i=0}^{n} y_i \right)$ is defined as the mass center of the network. The network center is calculated first, and other features depend on it.

Centrality:

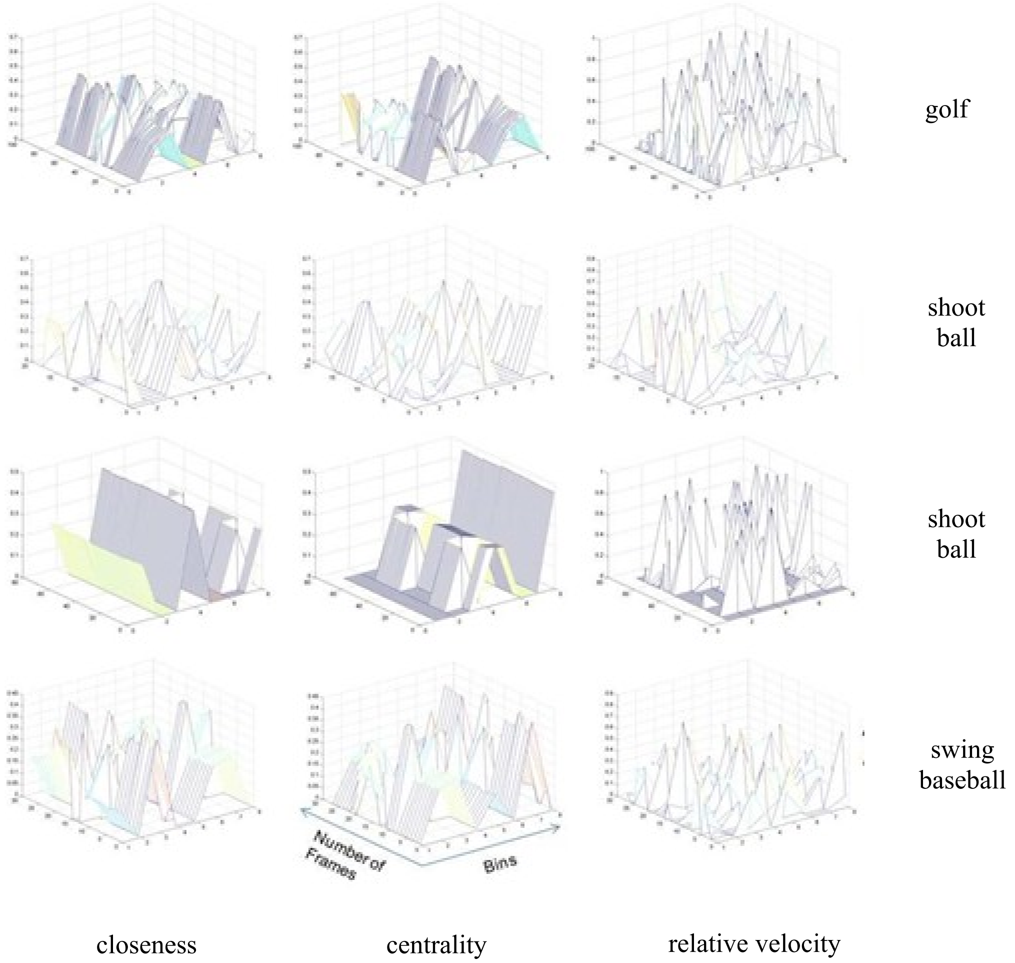
In general, centrality measures how the central node related to all other nodes in a social network. In our framework, centrality is used as a distance measurement between each node and the mess center of the network. Each node has a position $m_i = (x_i, y_i), (i = 1, \dots, n)$ in the network and the relative position to the network center is a directional vector $ce_i = \overrightarrow{m_i m_c}$. The centrality vector is designed as an 8-bin histogram of direction accumulating the magnitude of the distance and it is normalized. The centrality vector is written as $\mathbf{Ce}_t = \{ce_i\}_t, (i = 1, \dots, n; t = 8)$.

Closeness:

Closeness describes how close an individual is to all the rest nodes in a network. In our framework, the directional distance between each node to every other node in the network is calculated. Therefore, the distances of every pair of nodes $cl_{i,j} = \overrightarrow{m_i m_j}$ are accumulated in the closeness vector which is also a histogram with 8 bins of directions. It is denoted as $\mathbf{Cl}_t = \{cl_{i,j}\}_t, (i, j = 1, \ldots, n; i \neq j; t = 8)$.

Centrality with relative velocity:
Besides of the positions of each node in each frame of the sequence, we also consider the position difference v_i of each node in two consecutive frames. Similarly, the magnitude of the velocity is accumulated into orientation histograms and normalized at each direction. This is called the centrality with relative velocity, $\mathbf{V}_t = \{v_i\}_t (i = 1, \ldots, n;\ \ t = 8)$.

Following these definitions, a set of social network analysis based features extracted at each frame will form an SNA feature vector with 26 dimensions, including network center, centrality, closeness and centrality with relative velocity. A feature vector is calculated at each desired frame and as one entry in the feature matrix. A sequence with N frames will produce a SNA feature set in the dimension of 26 × N. Figure 3.23. shows examples of social network features from four interaction sequences, i.e. golf, shoot ball, shoot gun, and swing baseball respectively.



golf

shoot ball

shoot ball

swing baseball

closeness

centrality

relative velocity

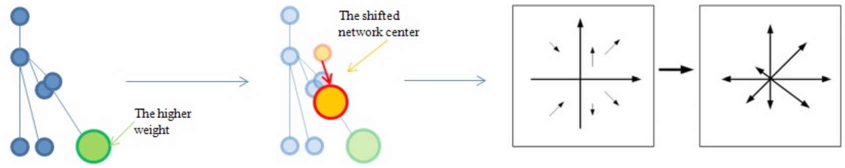
Figure 3.23. Examples of social network analysis based features on interactions.

3.3.4. Weighted Social network

Each node has its contribution in terms of forming a dynamic social network, and some may play more important roles than others. Therefore, the centrality weight is introduced to measure the influence of a node in the network, as shown in Figure 3.24.

Centrality weights: It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node than connections to low-scoring nodes. In the social network that describes human and object interaction, there are certain rules should be taken into consideration while assigning the weights of the nodes.

- The total weight of the network is normalized as one. $W_{hum} + W_{obj} = 1$, where $W_{hum} = \sum_{i}^{N_{hum}} w_i$ and $W_{obj} = \sum_{j}^{N_{obj}} w_j$.
- As human has more complicated structures and poses, there are more nodes on describing human than what on objects. $N_{hum} \geq N_{obj}$ and $W_{hum} \geq W_{obj}$.
- The objects have more important roles in understanding the interactions with human. Therefore each node on object has higher score than each node on human. $W_{hum} \geq W_{obj}$ and $w_i^{hum} \leq w_i^{obj}$.



A certain node (object) has a heavier weight than other nodes.

The mess center will be shifted.

Feature distribution: each feature is a histogram with 8 bins.

Figure 3.24. Example of SNA based features on human-object interactions. In a weighted social network, the network center shifted due to the unequal weighted nodes.

3.3.5. Experimental Results

We validate our method on HMDB dataset, which has 51 actions in five general types, and human motion with object interactions is one of them. Videos in this dataset are collected from various real world sources, like movies or YouTube. The video quality varies significantly, which makes the recognition task difficult.

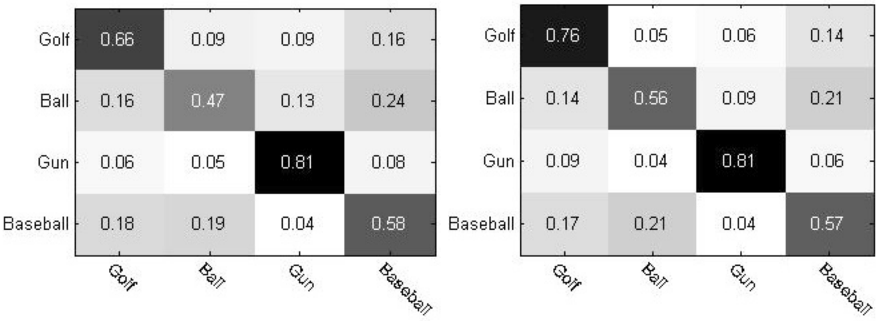
In our experiments, we choose four classes of interactions: swing golf club, shoot basketball, shoot gun, and swing baseball bat. Each class has 100 clips. We apply body parts and object detectors on every five frames in each sequence, and then extract the social network features from the detection results. In each activity class, there are four nodes representing human bodies, which are head, upper-body center and both hands, and one more node as the object.

In the classification stage, we apply two classifiers, SVM and HMM. Data clips contain different number of frames, and each frame is represented in a feature vector of 26 dimensions.

In the SVM approach, social network analysis based features from all frames are clustered and normalized before applying SVM. In our experiment, SVM with linear kernel is adopted and the training and testing data is divided into 50/50 with five-fold cross-validation. The classification results by SVM are shown in the confusion matrix in Figure 3.25.

In the HMM approach, we project the social network features into hidden Markov models with two hidden states and each state with two mixtures of Gaussian. The likelihood is computed between the test data and each trained HMM model, and the classification decisions are made according to the maximum likelihood. This experiment is also cross-validated for five times, and each time training and testing data is randomly divided into half and half. The classification results by HMM are shown in the confusion matrix in Figure 3.26.

The average classification accuracy is 63% and 67% by SVM classifier on SNA features and weighted SNA features respectively, and 71% and 74% by HMM. Some classes even have over 80% correct recognitions. From the results, we can observe that weighted SNA features outperform the un-weighted SNA features. The overall performance of our social network analysis based features is much higher than the benchmark [81] result by using the STIP features [52], which has accuracy around 20%.



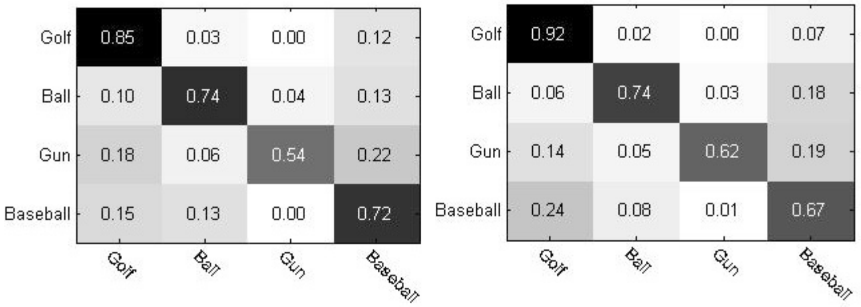
	Golf	Ball	Gun	Baseball
Golf	0.66	0.09	0.09	0.16
Ball	0.16	0.47	0.13	0.24
Gun	0.06	0.05	0.81	0.08
Baseball	0.18	0.19	0.04	0.58

(a) SNA

	Golf	Ball	Gun	Baseball
Golf	0.76	0.05	0.06	0.14
Ball	0.14	0.56	0.09	0.21
Gun	0.09	0.04	0.81	0.06
Baseball	0.17	0.21	0.04	0.57

(b) weighted SNA

Figure 3.25. The confusion matrix of SVM classification results on SNA and weighted SNA features.



(a) SNA (b) weighted SNA

Figure 3.26. The confusion matrix of HMM classification results on SNA and weighted SNA features.

3.4. Human Gait Recognition Based on Pyramid Histogram of Oriented Gradients (pHOG)

3.4.1. Introduction

During the development of SIRF, we have also studied various CL primitive exactors. One of these efforts was to develop a detector of human objects over a distance in low quality video sequence, when the object can only be recognized by its gait instead of its texture signatures. Such tool is very necessary in the construction of high level descriptions of small human group events.

Human gait recognition, as to identify different people from their silhouette of walking styles, has been studied for many years in computer vision community. Gait recognition can be used to identify people at a distance without a high resolution image, which could be a great advantage to other biometric identification approaches such as face recognition. However, gait recognition remains to be a challenging problem due to the variation of human appearance, different of viewing angle and scale.

Most previous works have been focused on the gait cycle modeling to identify different gait styles. To capture the dynamic uniqueness of each person's walking style, different feature such as Energy Intensity [86], Moving Motion Silhouette Image (MMSI) [87], Gait Entropy Image (GEnI) [88] have been proposed. In addition, many classification methods, such as Hidden Markov Model (HMM) [89], hierarchical structural model [90], manifold learning [91] are utilized to increase the robustness and effectiveness in gait recognition.

Capturing the dynamics of human silhouette is the key element in gait recognition. HOG feature [92] has been proven as a robust feature for pedestrian detection. As a local description feature, HOG feature captures the detailed gradient variation of small image patches, which inspired us to adapt HOG feature to describe the variation of human silhouette during a walking cycle. In this study, we propose a Pyramid HOG (pHOG) feature based human gait recognition framework.

To extract human silhouette, background subtraction is applied on the human gait images. The human silhouette images are then segmented into binary images and normalized to the same size. Instead of computing HOG at one scale, the HOG feature at different scale will be calculated and a pyramid for human silhouette representation will be constructed. The extracted pHOG features for each person will be used to train a Gaussian Mixture Model (GMM) based HMM for modeling the person’s gait cycle.

Compared with other features for gait recognition, the proposed pHOG feature on the binary silhouette image can effectively capture the shape characteristic of each person. As the human’s silhouette rather than other texture features plays an essential role in gait recognition, the binary image representation accurately characterizes the human body movement during each gait cycle, while the Pyramid of HOG provides an effective spatial encoding approach to represent the human body shapes at different scale. Beside its descriptiveness, the proposed novel feature also reduces the computational cost for motion features between frames.

3.4.2. Framework Overview

The proposed Human gait recognition framework can be divided to three parts: human silhouette extraction, pHOG feature computation and HMM model training and testing.

Human silhouette extraction

Given an input video, the silhouette extraction process is shown in Figure 3.27 below. First, the background image is assumed to be known from the video as shown in Figure 3.27(a). Following the background subtraction approach in [93], the raw silhouette images are obtained at this step, by subtracting the background image from the original frame. Then all the raw silhouette images will be refined in the shadow and noise removal process.

Background subtraction and Pixel classification

Each sequence consists of one person and one sequential movement. The background scene is obtained from the beginning of each sequence.

The raw images of human silhouette are then obtained by subtracting the original image with the selected background image. Due to illumination change or many other issues, there existed many noise in these raw images, such as human shadow and isolated parts, which are needed to be processed in next steps.

Each pixel in the background image is modeled by $< E_i, s_i, a_i, b_i >$. The arithmetic mean and standard deviation is calculated. E_i is expected color value of pixel *i* in the background image for R, G, B channel, i.e. $E_i = [\mu_R(i), \mu_G(i), \mu_B(i)]$; and s_i is the standard deviation $s_i = [\sigma_R(i), \sigma_G(i), \sigma_B(i)]$; a_i is variation of brightness distortion and b_i is variation of chromaticity distortion of the i^{th} pixel.

Then the background image is normalized at each R, G, B channel and horizontally aligned. Each image will be decomposed into brightness and chromaticity components. The measurement of brightness distortion (α_i) and chromaticity distortion (CD_i), which will be used for classifying each pixel are defined as below:

$$\alpha_i = \frac{\left(\frac{I_R(i)\mu_R(i)}{\sigma_R^2(i)} + \frac{I_G(i)\mu_G(i)}{\sigma_G^2(i)} + \frac{I_B(i)\mu_B(i)}{\sigma_B^2(i)} \right)}{\left(\left[\frac{\mu_R(i)}{\sigma_R(i)}\right]^2 + \left[\frac{\mu_G(i)}{\sigma_G(i)}\right]^2 + \left[\frac{\mu_B(i)}{\sigma_B(i)}\right]^2 \right)} \tag{3.27}$$

$$CD_i = \sqrt{\sum_{C=R,G,B} \left(\frac{I_C(i) - \alpha_i \mu_C(i)}{\sigma_C(i)} \right)^2} \tag{3.28}$$

In the next step, the variation of the brightness distortion a_i and chromaticity distortion b_i on i^{th} pixel is shown as below:

$$a_i = \sqrt{\frac{\sum_{i=0}^{N} (\alpha_i - 1)^2}{N}} \tag{3.29}$$

$$b_i = \sqrt{\frac{\sum_{i=0}^{N} (CD_i)^2}{N}} \tag{3.30}$$

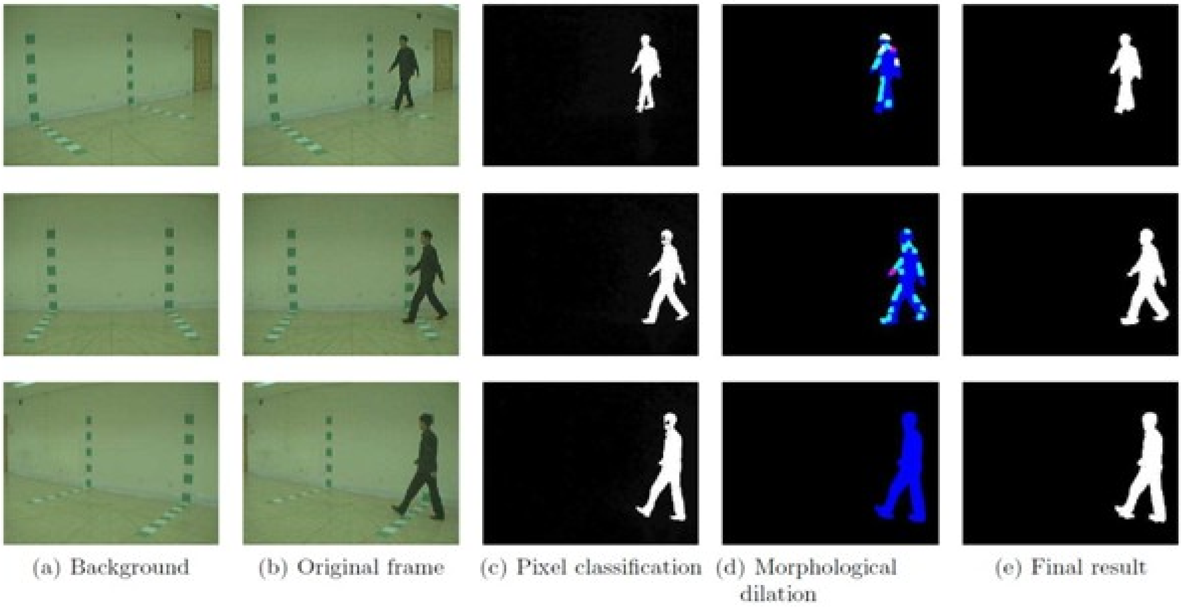

(a) Background (b) Original frame (c) Pixel classification (d) Morphological dilation (e) Final result

Figure 3.27. Human silhouette image extraction procedure

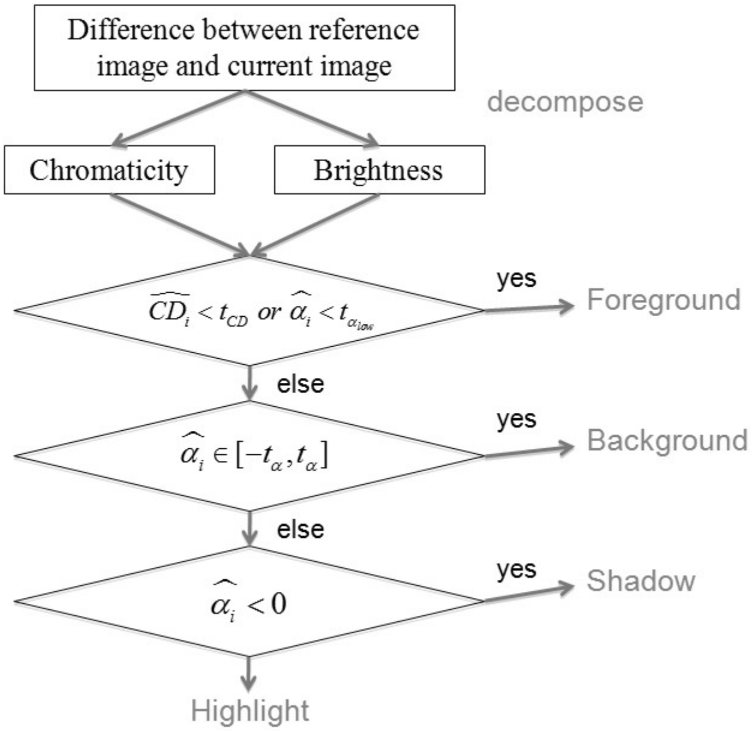

Difference between reference image and current image
decompose
Chromaticity
Brightness
$\widehat{CD_i} < t_{CD}$ or $\widehat{\alpha_i} < t_{\alpha_{low}}$
yes
Foreground
else
$\widehat{\alpha_i} \in [-t_\alpha, t_\alpha]$
yes
Background
else
$\widehat{\alpha_i} < 0$
yes
Shadow
Highlight

Figure 3.28. Pixel classification flow chart

In the last step, the pixel classification is performed. As shown in Figure 3.28, each pixel is classified into foreground, background, shadow or highlight by comparing the normalized brightness distortion $\widehat{\alpha_i} = \frac{\alpha_i - 1}{\alpha_i}$ and normalized chromaticity distortion $\widehat{CD_i} = \frac{CD_i}{b_i}$ with the thresholds. Some examples of background subtraction results are show Figure 3.27.

Noise removal and silhouette extraction

After pixel classification, there are still holes or some isolated parts in human silhouette images. To connect the isolated parts in the silhouette image, morphological operation is applied. After mathematical subtraction the background image and foreground image with respect to the shadow, the human silhouette image is reconstructed with morphological operation dilation.

After morphological operation, all the images are normalized to the same size for feature extraction. A bounding box (size 150×200) is used to crop the human silhouette in each image. All the images with small size silhouettes will be discarded. Finally, the size normalize and horizontal alignment within bounding box will be performed and and only the silhouette images within the bounding box only will be kept.

Some of the extracted silhouette images are shown in the Figure 3.29.

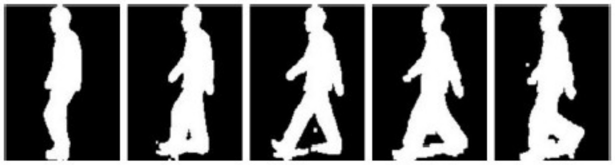
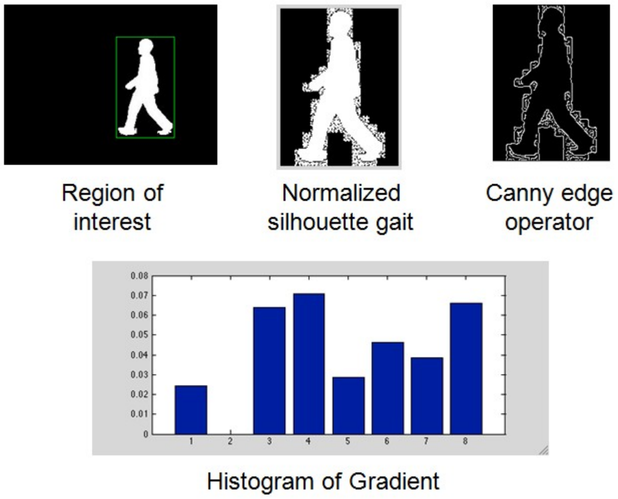


Figure 3.29. Silhouette examples

pHOG feature computation

HOG (Histogram of Oriented Gradient) is derived from SIFT feature. To extract HOG features, the whole detection window is firstly divided into grid of small cells. Similar to the cells for computing SIFT feature, the gradient in each cell is also accumulated at different directions in a histogram and form a normalized vector description. Pyramid HOG is an extension of HOG feature and it consists of HOG features at different levels with different block sizes. The pyramid HOG feature is utilized to describe the silhouette on all the binary images. In our experiment, the pyramid has two levels for each silhouette image and each block has 8 bins for directions, the total size of pHOG description is 648 for each binary silhouette image.



Region of interest

Normalized silhouette gait

Canny edge operator

Histogram of Gradient

Figure 3.30. Calculating HOG of silhouette gait

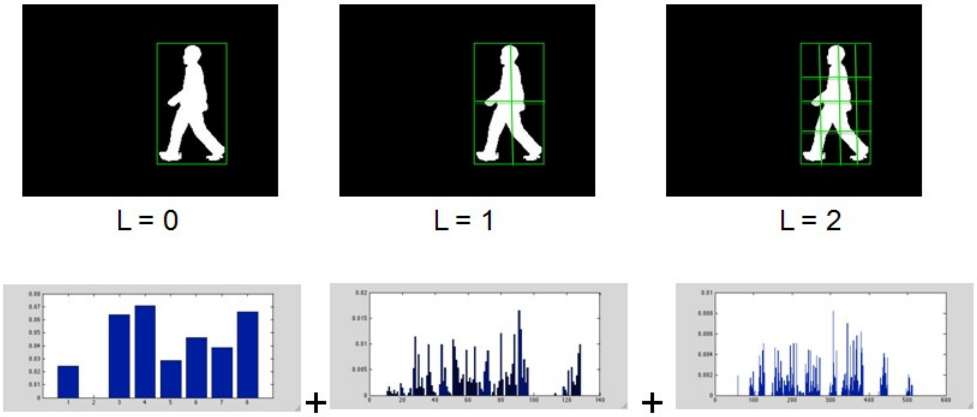


Figure 3.31. Calculating pHOG of silhouette gait. pHOG feature is accumulation of histograms at several pyramid levels. In our experiments, we use three levels of pyramid and a pHOG feature is a 648-dimension of histogram.

HMM based classification

To classify the extracted pHOG feature, a Gaussian Mixture Model (GMM) based Hidden Markov Model (HMM) is applied for gait recognition. The pHOG feature sequence of each subject is used to train a specific HMM model. Given a test gait sequence with pHOG feature, its likelihood probability with regard to all trained HMMs will be calculated and the one with the maximum likelihood will be selected as the classification output.

3.4.3. Experimental Results

To evaluate the proposed framework, we conduct gait recognition experiments on the CASIA Gait Dataset B [94]. In our experiments, we are using 31 subjects with 11 angles from $0^\circ \sim 180^\circ$ and each has 6 sequences. Sequences include $6 \sim 8$ walking cycles. Each sequence is divided into clips which contain one walking cycle in each.

Hidden Markov Models are used for classification in our framework, which are set to have three Mixture of Gaussian in each state. The recognition rate varies with the different number of states, as we can observe from Figure 3.32. The accuracies are very high and can even reach 100% in some cases. Accuracy is little less at 180° angle, which is the back view of a subject. The overall accuracy is shown in the Table 3.3. When Hidden Markov Model has 5 states, our proposed framework has the best results. Results are based on 5-fold cross validation.

Table 3.3. Gait recognition with Different number of states

HMM state number	Recognition rate
3	93.55%
4	94.84%
5	95.81%
6	94.51%
7	91.61%

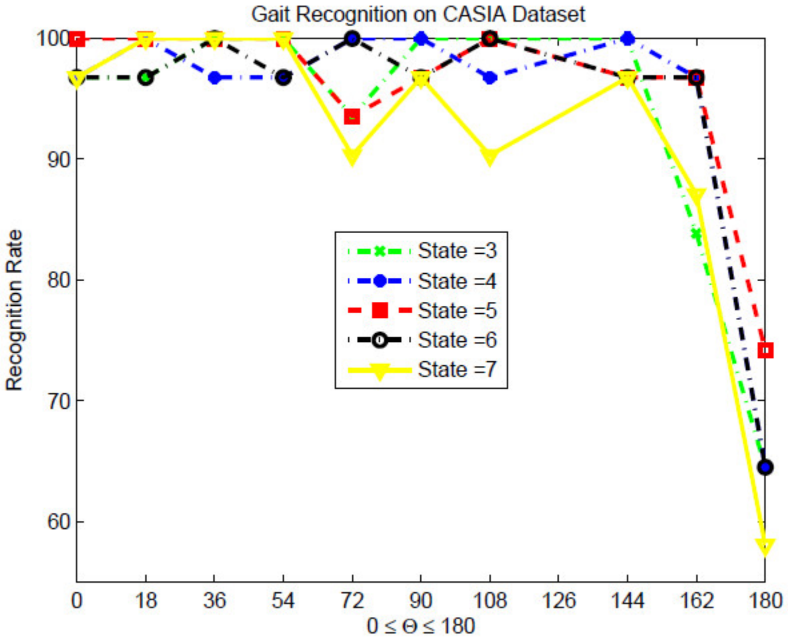


Figure 3.32. The effect of HMM state number on gait recognition performance

As shown in the Table 3.4, while HMM with 5 states and 3 mixture of Gaussian, our proposed framework can achieved 95.81% recognition rate. Compared to MMSI method: 73.00% and GEI method: 63.33% under the same experimental setting, our proposed framework demonstrated clear superior performance in human gait recognition.

Table 3.4. Gait recognition performance

	pHOG (our method)	MMSI [87]	GEI [86]
CASIA Dataset	95.33%	73.00%	63.33%

[87] Nizami, I. F., Hong, S., Lee, H., Lee, B., and Kim, E., "Automatic gait recognition based on probabilistic approach," *Int. J. Imaging Syst. Technol*. 20, 400–408 (Dec. 2010).
[86] Han, J. and Bhanu, B., "Individual recognition using gait energy image," *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (Feb. 2006).

4. Quantitative Metrics of Semantic Utility

4.1. Feature Selection via Sparse Imputation (FSSI)

4.1.1 Introduction

Within a multi-modality sensor network, sensor inputs at particular time instance can be conceptually aggregated into a single observation vector. Evaluating the utility of a particular sensor input is therefore equivalent to feature selection.

Feature selection is an important technique in machine learning and data mining. How to select the most useful features is a key factor in many applications, such as pattern recognition [95] and computer vision [96]. Functionally, feature selection can be divided into two groups: filter model [97] and wrapper model [98]. Filter model utilized different metrics to evaluate the individual feature, and remove some features before the prediction process. In wrapper method, the prediction results (or change of the results) of a model are used to measure the value of a feature. The computation cost limits the application of wrapper model on large data sets.

Filter is the most popular model in recent research, as it has low computational cost and is robust in theoretic analysis. Depends on the class labels, feature selection can be implemented in supervised fashion or unsupervised fashion. Most existing filter models are in supervised fashion. In real world applications, the class labels are always scarce. It is meaningful to design a filter feature selection method in unsupervised fashion.

How to design a meaningful evaluation metrics is the key for a good filter model. There are various metrics to build filters. Normally the metrics are kinds of relationship between the features and labels. Two popular filter metrics are mutual information [97] and correlation [99]. Max-relevance and min-redundancy feature subset are realized based the mutual information of the training data. Correlation based feature selection method, which is simple and fast to execute, is successfully applied on continuous class problems. There are also other effective filter metrics in recent researches. Class separability [100] is applied in a high dimensional kernel model and feature selection is carried on to maximize the separability. Error probability is considered as discriminating power [101], and it has been utilized to design feature selection.

Regarding the labeled training data and unlabeled training data. Feature selection can also be grouped as supervised feature selection and unsupervised feature selection. Supervised feature selection evaluates the relationship between the feature values and the label values. Fisher score [102] ranks the discriminative ability of individual feature according the labels. It is a simple and effective feature selection method. Unsupervised feature selection measures the feature similarity or local information. Laplacian score [103] evaluates the geometrical properties in the feature sets, which is an efficient unsupervised feature selection method.

Sparse coding (representation) [104] has been extensively studied in recent literature. It reconstructs a signal (data) through a linear combination of a minimum set of atom vectors from a dictionary. More specifically, a signal (data) $\mathbf{y} \in \mathbb{R}^m$ can be sparsely represented through $\mathbf{y} = \mathbf{Dx}$, with a well designed dictionary $\mathbf{D} \in \mathbb{R}^{m \times d}$. The correspondent coefficient vector $\mathbf{x} \in \mathbb{R}^d$ is sparse in the sense that most of its elements are zeros.

There are many successful applications with sparse representation, such as blind source separation [105], image denoising [106], sparse representation based classification [107] and sparse imputation [108]. Sparse imputation, which is introduced in [108] with application on speech recognition, is a new technique to use sparse representation to recover missing data.

In this section, we use sparse imputation to evaluate individual feature of training data, and the imputation quality of each feature is recorded to build a new feature selection filter model. This filter model is an unsupervised feature selection method, as the class labels don't contribute to sparse imputation. We use proposed feature selection method to apply on UCI data sets [109], and compared the classification performance with Fisher score method (supervised filter model) and Laplacian score method (unsupervised filter model). Comprehensive comparisons indicate the effectiveness of our method.

The main contributions of this method are summarized as follows:

- A new filter model Feature Selection via Sparse Imputation (FSSI) is presented. In particular, the imputation quality for individual feature is utilized as evaluation metrics in feature selection.

- The proposed method is applied to UCI data sets (binary-category and multiple-category). The classification results are obtained with classic classifiers (support vector machine, k nearest neighbors and multi-layer feed-forward networks).

- The proposed unsupervised feature selection filter model is compared with other methods, Fisher score method (supervised filter model) and Laplacian score method (unsupervised filter model). The comparison results on UCI data sets demonstrate the capability and efficiency of our method.

4.1.2. Related Work

We consider a data set $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \in \mathbb{R}^{n \times m}$, n is the total number and m is the dimension for each data. The labels of the data set are $C = \{c_1, c_2, \cdots, c_n\}$. The original feature sets are $\mathbf{F} = \{F_1, F_2, \cdots, F_m\}$, and the purpose of feature selection is to identify the feature subsets $\mathbf{F}_s = \{F_1, F_2, \cdots, F_s\}$ with $s < m$. Based on above setting, we introduce Fisher score method, Laplacian score method and sparse imputation method.

Fisher score

Fisher score [110] is one of the most popular **supervised** feature selection method. The key idea of Fisher score is to search for a feature subset, the distances for different classes in the feature subset are as large as possible, while the distances of the same class in the feature subset are as small as possible. Fisher score method has been improved recently. In [111], a clustering method specialized for Fisher score is developed, which is able to detect important dimensions. Generalized Fisher score is proposed in [112], which can optimize the lower bound of traditional Fisher score. The criterion of Fisher score is described in the following part.

Consider a data set $\{\mathbf{y}_i, c_i\}$, $c_i \in \{1, 2, \cdots, k\}$ and $i = 1, 2, \cdots, n$, where k is the number of the classes. Let n_j denote the number of data in class j and $j = 1, 2, \cdots, k$, so $n_1 + n_2 + \cdots + n_k = n$. For a feature set F_z in $\mathbf{Y}$, let μ and σ^2 denote mean and variance, and μ_j and σ_j^2 are the mean and variance for certain categorical data. According to [110], $\Sigma_{j=1}^{k} n_j \sigma_j^2$ is the within-class variance, and $\Sigma_{j=1}^{k} n_j (\mu_j - \mu)^2$ is the between-class variance. Fisher score S_z for feature F_z is calculated as:

$$S_z = \frac{\Sigma_{j=1}^{k} n_j (\mu_j - \mu)^2}{\Sigma_{j=1}^{k} n_j \sigma_j^2} \tag{4.1}$$

The score for individual feature is recorded based on above equation and it can contribute to feature selection.

Laplacian score

Laplacian score is proposed based on Laplacian Eigenmaps [113] and Locality Preserving Projection [114], the key idea is to evaluate the features through the features' locality preserving properties. It is a classical **unsupervised** filer model for feature selection. For the training data $\mathbf{Y}$ with feature set $\mathbf{F}$, let L_z is the Laplacian score for the z th feature F_z, the Laplacian score is computed as:

1. It first establishes a nearest neighbor graph G with diverse data nodes ($\mathbf{y}_\alpha$ and $\mathbf{y}_\beta$, $\alpha, \beta = 1, \cdots, n$) in the data set, define $G_{\alpha,\beta} = e^{-\frac{\|\mathbf{y}_\alpha - \mathbf{y}_\beta\|^2}{t}}$, where t is a predefined constant.

2. For each feature F_z, the feature values are $F_z = \mathbf{f_z}$, then L_z can be calculated as

$$L_z = \frac{\widetilde{\mathbf{f}_z^T} L \mathbf{f}_z}{\widetilde{\mathbf{f}_z^T} Q \mathbf{f}_z} \tag{4.2}$$

where $Q = diag(G\mathbf{1}), \mathbf{1} = [1, \cdots, 1]^T, L = Q - G$, and $\widetilde{\mathbf{f}_z}$ is a normalization through:

$$\widetilde{\mathbf{f}_z} = \mathbf{f}_z - \frac{\mathbf{f}_z^T Q \mathbf{f}_z}{\mathbf{1}^T Q \mathbf{1}} \tag{4.3}$$

Similar as Fisher score, Laplacian score based on Eq. (4.2) are recorded for feature selection.

Sparse imputation

Assume a data set $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_r, \cdots, \mathbf{y}_n\} \in \mathbb{R}^{n \times m}$ and a suitable dictionary $\mathbf{D} \in \mathbb{R}^{d \times m}$ for the sparse coding, where $d > m$ ensures over-complete basis vectors (or atoms). We seek **x** such that **Dx**=**y**. The original sparsity constrain on **x** is defined in l_0 norm:

$$min \| \mathbf{x} \|_0, s.t. \mathbf{Dx} = \mathbf{y} \tag{4.4}$$

However, the solution of above equation is NP hard. According to Restricted Isometry Property (RIP) [115], the l_0 optimization could be relaxed to l_1 optimization. And the sparse coding could be expressed as:

$$min \| \mathbf{x} \|_1, s.t. \mathbf{Dx} = \mathbf{y} \tag{4.5}$$

The l_1 optimization is a convex optimization problem. Considering reconstruction errors, a more practical ℓ_1-regularized least squares optimization [116] is formulated as:

$$\hat{\mathbf{x}} = arg\min\{\| \mathbf{y} - \mathbf{Dx} \|_2^2 + \lambda \| \mathbf{x} \|_1\} \tag{4.6}$$

There are also other efficient methods to do the sparse coding, such as matching pursuit [117] and basis pursuit [118]. The dictionary in sparse coding is important, and extensive research works have contributed many effective dictionary learning methods, such as online dictionary learning [119] and Laplacian score dictionary [120]. As the focus of this work is on sparse representation for imputation, we just use all the training data to build the dictionary, which is similar as sparse representation based classification [107].

Imputation [121] is a statistic method for handling missing data. The sparse coding framework could be transferred to handle imputation. In particular, suppose that a data set $\mathbf{Y} \in \mathbb{R}^{n \times m}$ contains $\mathbf{Y_r} \in \mathbb{R}^{n \times q}$ (reliable feature subsets) and $\mathbf{Y_u} \in \mathbb{R}^{n \times (m-q)}$ (unreliable feature subsets):

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y_r} \\ \mathbf{Y_u} \end{bmatrix}$$

accordingly the dictionary $\mathbf{D} \in \mathbb{R}^{d \times m}$ contains $\mathbf{D_r} \in \mathbb{R}^{d \times q}$ and $\mathbf{D_u} \in \mathbb{R}^{d \times (m-q)}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_r} \\ \mathbf{A_u} \end{bmatrix}$$

The sparse coding process is carried out on the reliable feature subsets:

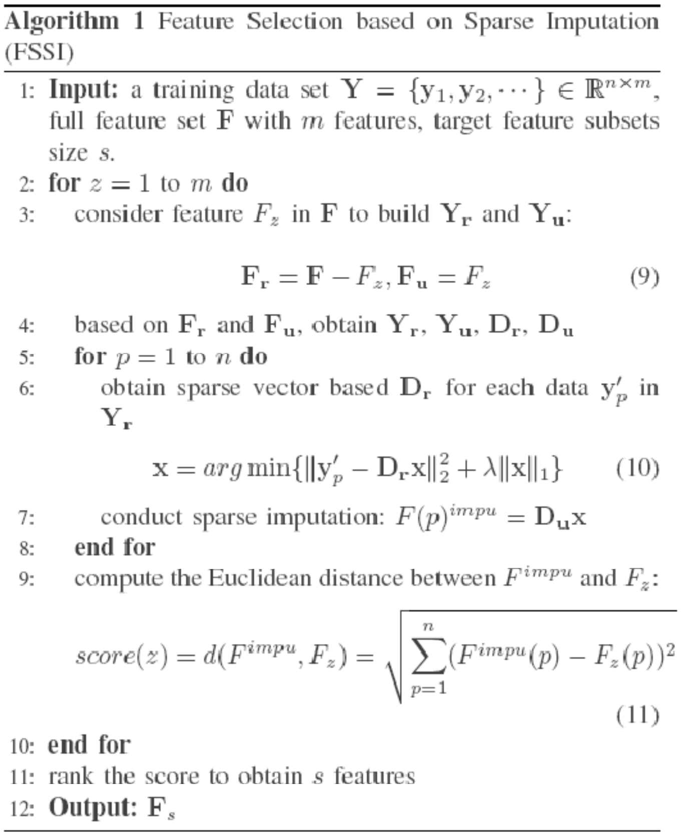
$$\mathbf{x} = arg \min\{\| \mathbf{y_r} - \mathbf{A_r x} \|_2^2 + \lambda \| \mathbf{x} \|_1\} \tag{4.7}$$

Then use the sparse vector to apply on the unreliable dictionary $\mathbf{D_u}$ to realize imputation, which is $\mathbf{y}_u = \mathbf{D_u x}$. And sparse imputation can be described as:

$$\hat{\mathbf{y}} = \begin{cases} \widehat{\mathbf{y}_r} = \mathbf{y}_r \\ \widehat{\mathbf{y}_u} = \mathbf{A_u x} \end{cases} \tag{4.8}$$

4.1.3. FSSI METHOD

In this section, we introduce the Feature Selection via Sparse Imputation (FSSI) method. The key idea is to use the sparse imputation to recover each feature, and then the quality of the representation is the criterion to do the feature selection. The details process is shown in Algorithm 1.



Algorithm 1 Feature Selection based on Sparse Imputation (FSSI)

```
1: Input: a training data set $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots\} \in \mathbb{R}^{n \times m}$,
   full feature set $\mathbf{F}$ with $m$ features, target feature subsets
   size $s$.
2: for $z = 1$ to $m$ do
3:   consider feature $F_z$ in $\mathbf{F}$ to build $\mathbf{Y_r}$ and $\mathbf{Y_u}$:
         $\mathbf{F_r} = \mathbf{F} - F_z, \mathbf{F_u} = F_z \qquad (9)$
4:   based on $\mathbf{F_r}$ and $\mathbf{F_u}$, obtain $\mathbf{Y_r}$, $\mathbf{Y_u}$, $\mathbf{D_r}$, $\mathbf{D_u}$
5:   for $p = 1$ to $n$ do
6:     obtain sparse vector based $\mathbf{D_r}$ for each data $\mathbf{y}'_p$ in
       $\mathbf{Y_r}$
         $\mathbf{x} = arg \min\{\|\mathbf{y}'_p - \mathbf{D_r}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1\} \qquad (10)$
7:     conduct sparse imputation: $F(p)^{impu} = \mathbf{D_u}\mathbf{x}$
8:   end for
9:   compute the Euclidean distance between $F^{impu}$ and $F_z$:
         $score(z) = d(F^{impu}, F_z) = \sqrt{\sum_{p=1}^{n}(F^{impu}(p) - F_z(p))^2} \qquad (11)$
10: end for
11: rank the score to obtain $s$ features
12: Output: $\mathbf{F}_s$
```

In this method, we use sparse imputation to evaluate each feature. In details, we first treat a feature set F_z as an unreliable feature sets $\mathbf{F}_u$. Then we establish reliable feature sets $\mathbf{F}_r$, which is removed F_z from the whole feature sets $\mathbf{F}$. The data sets $\mathbf{Y_r}$ and $\mathbf{Y_u}$ are based on feature setting of $\mathbf{F}_r$ and $\mathbf{F}_u$. In the process of imputation, the dictionary $\mathbf{D}$ are the whole training data [107]. Therefore $\mathbf{Y_r} = \mathbf{D_r}$ and $\mathbf{Y_u} = \mathbf{D_u}$ in our method. Based on sparse coding equation at step 6, the sparse vector $\mathbf{x}$ for each data $\mathbf{y}'_p$ in $\mathbf{Y_r}$ are calculated ($\mathbf{y}'_p$ is based on the feature set of $\mathbf{Y_r}$). At step 7, the imputation of unreliable feature for data $\mathbf{y}'_p$ is conducted. Then based on equation at step 9, sparse imputation quality of the feature set F_k is computed. Finally, step 11 ranks the features based on imputation quality from the worst to the best, and output the feature subsets with target dimensions.

Our FSSI method is proposed based on two perspectives:

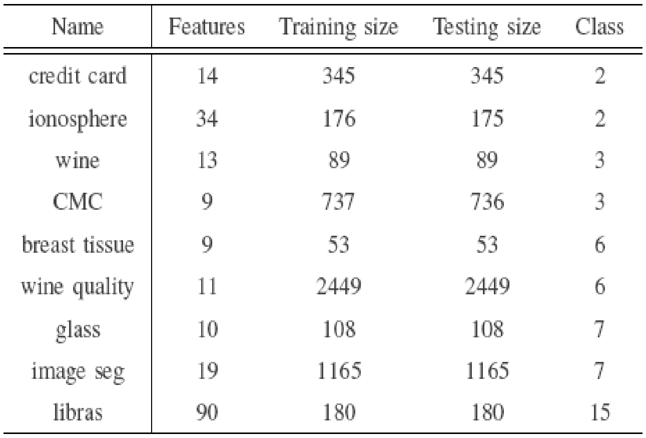
- FSSI method shares the prosperities of AdaBoost [122]. In the loop of Adaboost, the misclassified training data would be increased the weights in the next iteration. In our method, the feature with the worst imputation quality would be rank first in feature selection. There are successful feature selection method based Adaboost. In [123], AdaBoost is efficiently used to select the global and local appearance features for face recognition. Adaboost has applied on selecting Gabor Feature for image classification [124], and results are competitive with low memory and computation cost.

- Sparse imputation aims to represent a data (or feature) with existing dictionary. When a data (or feature) has idea sparse imputation quality, it may conclude that the existing dictionary contains the information of the data. Therefore the data is not necessary for the learning system considering memory and computation factors. Whereas a data (or feature) has unacceptable sparse imputation based on existing dictionary, the data should be added in the learning system to improve the diversity.

4.1.4. Experimental Results

In this section, the empirical studies are conducted on the nine data sets from UCI Repository [109] to show effectiveness of FSSI method. There are two binary data sets and seven multiple categorical data sets. We focus on multiple classifications based on two factors: (1) compared with binary classification, multiple classifications are rare in research. (2) The performance of multiple classifications needs improvement in a lot of applications. The details information of the experimental data sets are shown in Table 4.1. Data "credit card" and data "ionosphere" are binary data sets. The rest are multiple categorical data sets. "CMC" is the abbreviation of "Contraceptive Method Choice", and "image seg" is the abbreviation of "Statlog (Image Segmentation)".

Table 4.1. UCI and face experiment datasets



Name	Features	Training size	Testing size	Class
credit card	14	345	345	2
ionosphere	34	176	175	2
wine	13	89	89	3
CMC	9	737	736	3
breast tissue	9	53	53	6
wine quality	11	2449	2449	6
glass	10	108	108	7
image seg	19	1165	1165	7
libras	90	180	180	15

Configurations

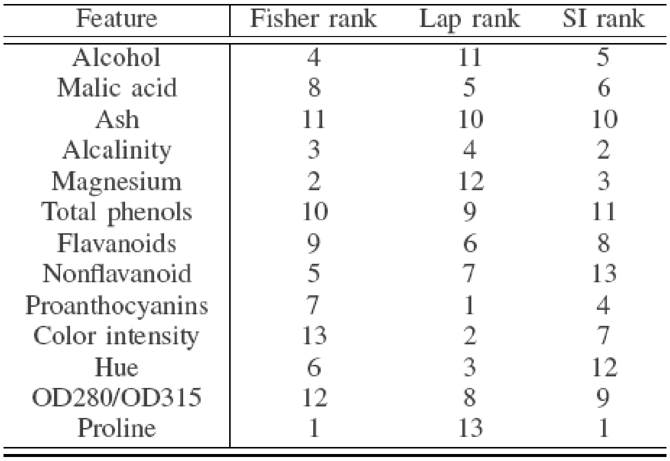
In the experiment, each data set is randomly separated into two equal parts, in which one part are training data and the other part are testing data. The training data are used to establish the model for feature selection. Three filter feature selection methods are utilized in the experiment for comparison: Fisher score (supervised feature selection method), Laplacian score (unsupervised feature selection method) and our proposed feature selection via sparse imputation method. We use Fisher, Lap and FSSI as abbreviations to represent these three methods in the experiment.

We use feature-based classification as the evaluation criterion [125] to assess different feature selection methods. In particular, a feature selection operator Φ is defined as follows:

- The feature selection operator Φ is trained on the training data based on different algorithms, such as Fisher, Lap and FSSI.
- Update the data $\mathbf{Y}$ based on the operator Φ: $\mathbf{Y}' \leftarrow \Phi(\mathbf{Y})$
- Establish a classifier based the training part of $\mathbf{Y}'$ and record the classification performance on the testing part of $\mathbf{Y}'$

The experiment on each data set is conducted five times and average results are obtained. The target features size is from one to around 80% of whole feature size to show the comprehensive performances. Table 4.2 shows a case study on data wine. All 13 features are listed in the table. Fisher, Lap and sparse imputation (SI) have been trained on the training data to rank the features respectively. In particular, when the target feature size is 2, Fisher method would select "Proline and Magnesium" features for classification, Lap method would choose "Proanthocyanins and Color intensity" features, and SI method would select "Proline and Alcalinity" features.

Table 4.2. Feature selection on data wine



Feature	Fisher rank	Lap rank	SI rank
Alcohol	4	11	5
Malic acid	8	5	6
Ash	11	10	10
Alcalinity	3	4	2
Magnesium	2	12	3
Total phenols	10	9	11
Flavanoids	9	6	8
Nonflavanoid	5	7	13
Proanthocyanins	7	1	4
Color intensity	13	2	7
Hue	6	3	12
OD280/OD315	12	8	9
Proline	1	13	1

Three classic classifiers are used in our experiment: *k* nearest neighbor (*k* = 5 in the experiment), LibSVM [126] and multi-layer feed-forward networks [127]. We abbreviate above classifiers as 5-NN, LibSVM and NeuralNet. "1-v-r" method [128] is utilized when LibSVM and NerualNet handle multi-class data sets. The sparse coding toolbox is from [116]. Figure 4.1 shows an example of classification performance on data wine with different selected features. The outputs of FSSI are more accurate than that of Lap, we may claim that the feature selection method FSSI is more appropriate than Lap for data wine.

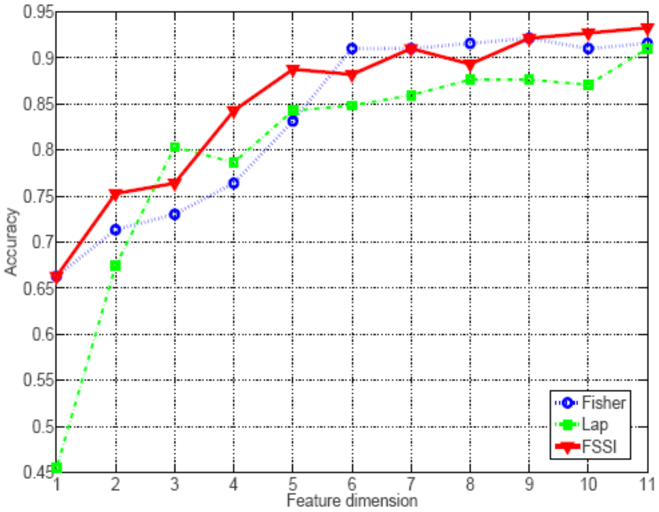


Figure 4.1. An example of feature selection based classification on data wine. The feature selection projector is trained with Fisher, Lap and FSSI respectively. Then updated training data and testing data are applied on 5-NN classifier to obtain the classification results on different dimensions. Data wine has 13 features, feature selection has targeted to 11 features, which is 84% of the whole feature sets.

Comparisons among FSSI, Fisher and Lap

In this section, the comparison results among FSSI, Fisher and Lap are shown. For briefness and clarity, we only show the plots of six data sets (two binary data sets and four multiple categorical data sets).

Figure 4.2 shows the comparison results for data credit. When the selected feature size is larger than 3, the advantage of FSSI could be observed with all three classifiers. It is also interesting to notice that the results of Fisher and Lap are almost same in the left two subfigures. However, the result of another binary data ionosphere from Figure 4.3 is unremarkable. Through the curves of FSSI are higher than that of Lap, the curves of FSSI are in the same level of Fisher.

When the experiments are carried on the multiple categorical data sets (Figure 4.4 to Figure 4.7), the effectiveness of FSSI could be noted. In Figure 4.4, the results of FSSI are encouraged. In the subfigure of “LibSVM”, the outputs of FSSI dominate the competitors from the dimension of 5. In the subfigures of “5-NN” and “NeuralNet”, the results of FSSI surpass other methods from the dimension of 2.

The advantages of FSSI also could be observed in Figure 4.5. for data breast tissue, in which the curves of FSSI always appear in the top of subfigures. Figure 4.6 shows the experimental results for data glass, Fisher and FSSI methods have improved results compared to Lap method. And in Figure 4.7 for data Libras, Lap and Fisher methods have enhanced outputs compared to Fisher method.

For intensive comparison of different feature selection methods, the statistic analysis is applied with experiment results. For each data set and each method, the classification accuracies, from feature size one to around 50% of whole feature sizes, are averaged and standard deviations are calculated [129] in the Table 4.3. The highest accuracy is highlighted. We can obverse that FSSI can win 5, 6 and 6 times in LibSVM, 5-NN and NeuralNet separately. The standard deviation of FSSI is slightly larger than competitors, which may indicate that the features selected by FSSI are contributable.

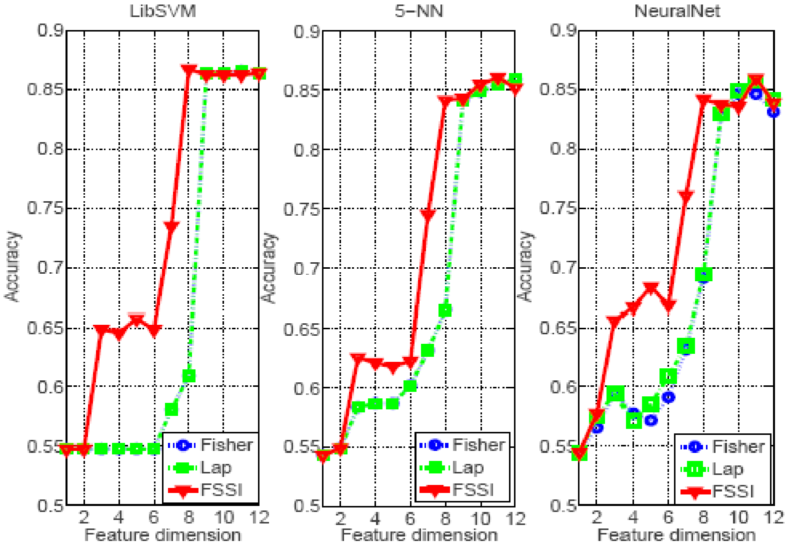


Figure 4.2. Comparison of feature selection based classification accuracies for data credit card. Feature selection methods Fisher, Lap and FSSI are used.(Left) Average accuracy of data credit card with LibSVM. (Center) Average accuracy of data credit card with 5-N (Right) Average accuracy of data credit card with NeuralNet.

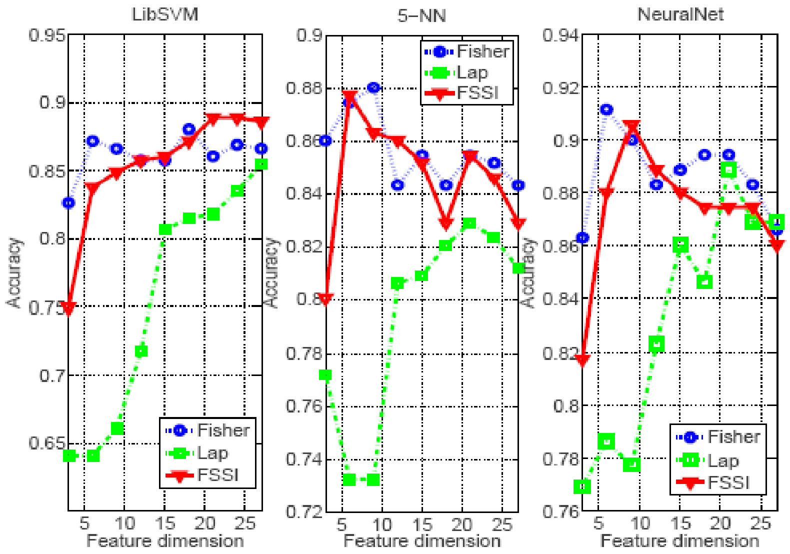


Figure 4.3. Comparison of feature selection based classification accuracies for data ionosphere. Feature selection methods Fisher, Lap and FSSI are used. (Left) Average accuracy of data ionosphere with LibSVM. (Center) Average accuracy of data ionosphere with 5-NN. (Right) Average accuracy of data ionosphere with NeuralNet.

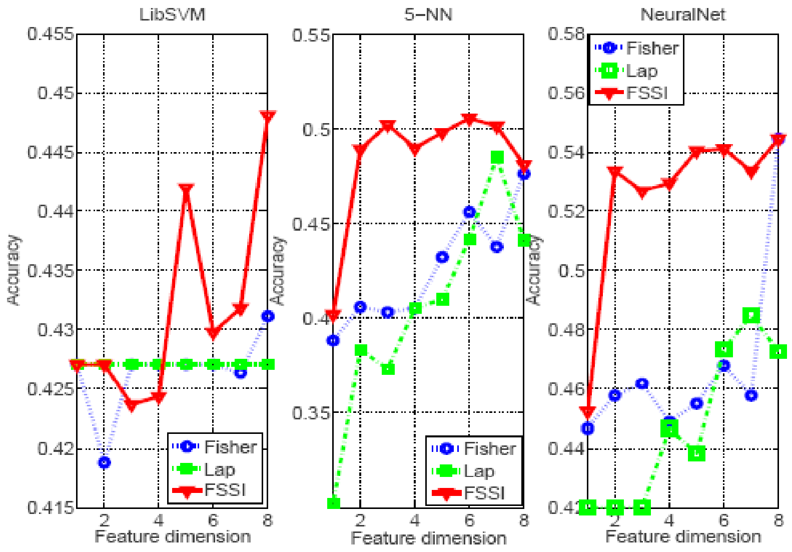


Figure 4.4. Comparison of feature selection based classification accuracies for data CMC. Feature selection methods Fisher, Lap and FSSI are used. (Left) Average accuracy of data CMC with LibSVM. (Center) Average accuracy of data CMC with 5-NN. (Right) Average accuracy of data CMC with NeuralNet.

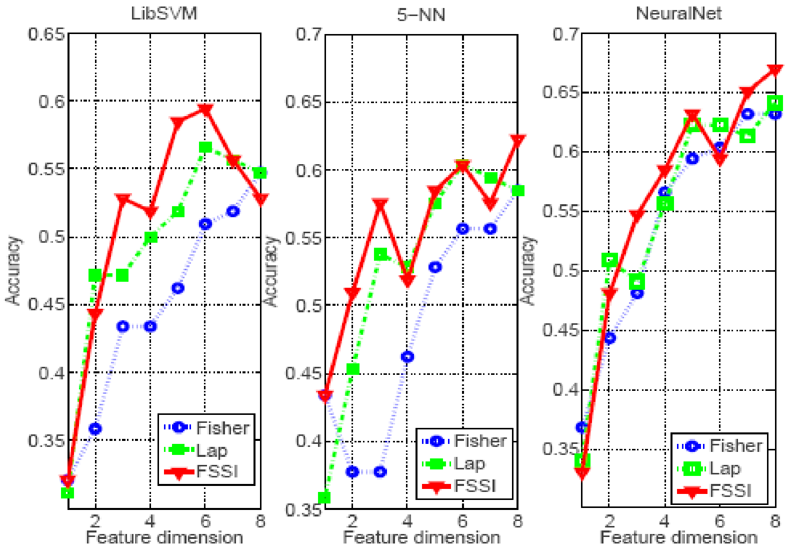


Figure 4.5. Comparison of feature selection based classification accuracies for data breast tissue. Feature selection methods Fisher, Lap and FSSI are used. (Left) Average accuracy of data breast tissue with LibSVM. (Center) Average accuracy of data breast tissue with 5-NN. (Right) Average accuracy of data breast tissue with NeuralNet.

Table 1.3. Accuracy average and standard deviation in low dimension

Data set	Evaluation	LibSVM			5-NN			NeuralNet		
		Fisher	Lap	FSSI	Fisher	Lap	FSSI	Fisher	Lap	FFSI
credit	Mean	0.552	0.552	**0.633**	0.583	0.583	**0.618**	0.582	0.588	**0.651**
	StDev	0.013	0.013	0.066	0.030	0.030	0.067	0.027	0.029	0.071
ionosphere	Mean	**0.856**	0.694	0.831	**0.863**	0.770	0.851	**0.889**	0.803	0.875
	StDev	0.018	0.070	0.046	0.015	0.038	0.030	0.018	0.038	0.034
wine	Mean	0.760	0.733	**0.791**	0.789	0.753	**0.815**	0.795	0.769	**0.806**
	StDev	0.089	0.207	0.086	0.097	0.146	0.091	0.063	0.174	0.054
CMC	Mean	0.425	0.427	**0.429**	0.407	0.375	**0.476**	0.454	0.429	**0.516**
	StDev	0.004	0	0.008	0.016	0.043	0.042	0.006	0.013	0.036
breast tissue	Mean	0.402	0.455	**0.479**	0.436	0.491	**0.525**	0.491	0.504	**0.515**
	StDev	0.060	0.083	0.102	0.063	0.086	0.061	0.092	0.105	0.117
wine quality	Mean	0.516	0.500	**0.524**	0.519	0.504	**0.528**	0.523	0.512	**0.533**
	StDev	0.042	0.021	0.043	0.057	0.027	0.057	0.047	0.019	0.049
glass	Mean	**0.762**	0.364	0.738	0.705	0.436	**0.709**	0.736	0.446	**0.743**
	StDev	0.037	0.021	0.029	0.007	0.069	0.011	0.015	0.070	0.022
image seg	Mean	**0.752**	0.494	0.569	**0.810**	0.573	0.668	**0.830**	0.597	0.706
	StDev	0.267	0.164	0.211	0.249	0.159	0.216	0.260	0.177	0.230
libras	Mean	0.411	**0.530**	0.515	0.463	**0.594**	0.588	0.590	**0.784**	0.742
	StDev	0.150	0.077	0.111	0.124	0.017	0.068	0.184	0.073	0.109
Mean Wins		3	1	5	2	1	6	2	1	6

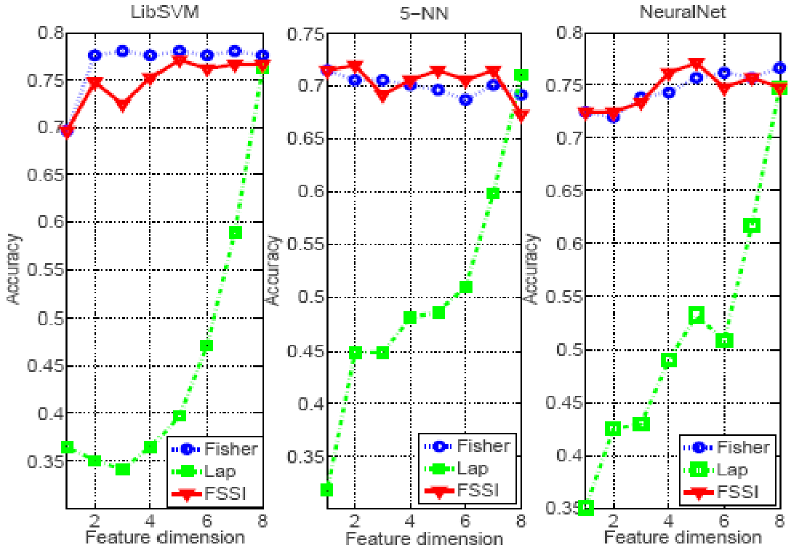


Figure 4.6. Comparison of feature selection based classification accuracies for data glass. Feature selection methods Fisher, Lap and FSSI are used.(Left) Average accuracy of data glass with LibSVM. (Center) Average accuracy of data glass with 5-NN. (Right) Average accuracy of data glass with NeuralNet.

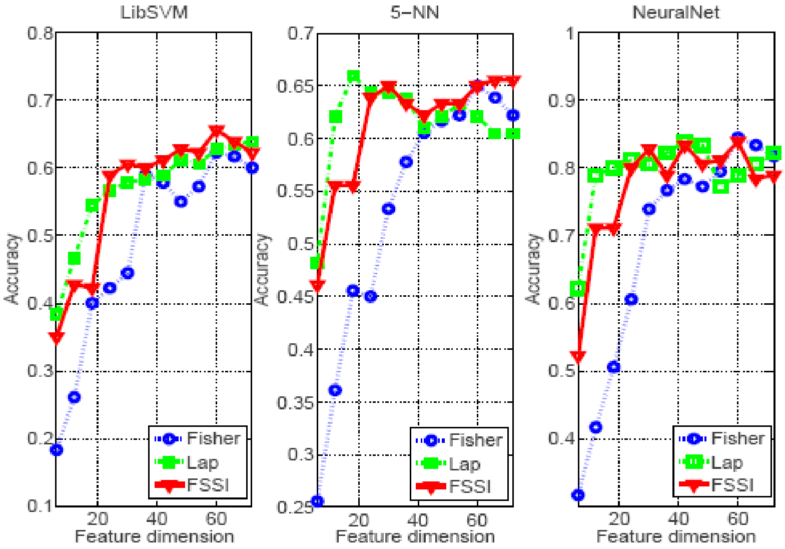


Figure 4.7. Comparison of feature selection based classification accuracies for data libras. Feature selection methods Fisher, Lap and FSSI are used. (Left) Average accuracy of data libras with LibSVM. (Center) Average accuracy of data libras with 5-NN. (Right) Average accuracy of data libras with NeuralNet.

4.2. ℓ_1Graph Based on Sparse Coding for Feature Selection

4.2.1. Introduction

We further extended our sparse coding based feature selection and utility score evaluation scheme into a more general method called $\ell_1$graph. In this work, we utilize the relations established by sparse coding between the signal (data) vector and various dictionary atoms to build ℓ_1 graphs. The graph has the properties in local preserving ability. We can therefore evaluate these properties to rank features and establish a new **unsupervised** filter model for feature selection.

The main contributions of this work are summarized as follows:

- A graph is established through ℓ_1-Norm Regularization. The linear relations between the signal and the dictionary atoms are shown on the graph.
- The features' local preserving ability is evaluated through spectral graph theory. The unsupervised filter model is established based on the ability to perform feature selection.
- The proposed method is applied to UCI benchmark data sets (binary-category and multiple-category). A 2-D visualization case study is carried out and compared with classic filter feature selection methods (Fisher score, Laplacian score and Pearson correlation [130]). Intensive experiments of feature based classifications are conducted to demonstrate the efficiency and effectiveness of our method.

4.2.2. Feature Score Based on ℓ_1Graph

We assume a data set **Y**={y$_1$, y$_2$, ...y$_i$, ..., y$_n$}$\in R^{n\times m}$. Our proposed method utilizes the property of *self-characterization* in the data sets. In detail, a data (signal) can be represented by other data from the same data set through

$$\mathbf{y}_i = \mathbf{Y}\mathbf{x}_i, \quad x_{ii} = \mathbf{0} \tag{4.9}$$

where **x**$_i$=[x_{i1}, x_{i2}, ..., x_{in}] and constraint x_{ii}=0 avoids the trivial solution of characterizing a data as a linear combination of itself. This formula is naturally transferred to sparse coding when we want to choose as less as possible data to represent **y$_i$** .We assume the dictionary of sparse coding is the whole data set. Then the constrain of ℓ_0 norm is:

$$\boldsymbol{min} \|\mathbf{x}_i\|_0, \ \boldsymbol{s.t.}\ \mathbf{y}_i = \mathbf{Y}\mathbf{x}_i \tag{4.10}$$

According to Restricted Isometry Property (RIP), the ℓ_0 norm can be transferred to ℓ_1 form and solved with ℓ_1-regularized least squares method [116]:

$$\mathbf{x}_{\mathrm{i}} = \boldsymbol{arg}\,\mathbf{min}\{\|\,\mathbf{y}_{\mathrm{i}} - \mathbf{Y}\mathbf{x}_{\mathrm{i}}\,\|_2^2 + \boldsymbol{\lambda}\,\|\,\mathbf{x}_{\mathrm{i}}\,\|_1\} \tag{4.11}$$

We summarize the sparse coding method for all data with the matrix form through:

$$\boldsymbol{min}\,\|\,\mathbf{X}\,\|_1, \quad \boldsymbol{s.t.} \quad \mathbf{Y} = \mathbf{YX}, \quad \boldsymbol{diag}(\mathbf{X}) = \mathbf{0} \tag{4.12}$$

Inspired by the work of sparse subspace clustering [131], the **similarity matrix** of ℓ_1 graph can be defined as:

$$\mathbf{W} = |\,\mathbf{X}\,| + |\,\mathbf{X}^{\boldsymbol{T}}\,|, \quad \boldsymbol{diag}(\mathbf{W}) = \mathbf{0} \tag{4.13}$$

which means a node (signal) *i* is connected with node *j* by an edge with the weight $|x_{ij}|+|x_{ji}|$. Based on the graph established above, our proposed feature score *S* based on the spectral graph theory [132] is computed as:

1. First, ℓ_1 graph **G** (**G**$_{ij}$ = **W**$_{ij}$) is built based on similarity matrix with nodes (signals) (**Y**={y$_1$, y$_2$, y$_3$, ...y$_i$, ..., y$_n$}).
2. For each feature F_z, the feature sets are F_z=**f**$_{\mathbf{z}}$, then S_z can be computed as

$$\boldsymbol{S}_z = \frac{\tilde{\mathbf{f}}_z^{\,T} \boldsymbol{L} \tilde{\mathbf{f}}_z}{\tilde{\mathbf{f}}_z^{\,T} \boldsymbol{Q} \tilde{\mathbf{f}}_z} \tag{4.14}$$

where $Q = diag(G1), 1 = [1, \cdots, 1]^T, L = Q - G$, and $\tilde{\mathbf{f}}_z$ is a classic normalization through:

$$\tilde{\mathbf{f}}_z = \mathbf{f}_z - \frac{\mathbf{f}_z^T \boldsymbol{Q}\mathbf{1}}{\mathbf{1}^T \boldsymbol{Q}\mathbf{1}}\mathbf{1} \tag{4.15}$$

The step 2 is based on the local property of each feature, G_{ij} evaluates the similarity between the *i*-th and *j*-th data (nodes). In detail, when two nodes have heavily weighted edge, the good feature should have close value between these two nodes. The heuristic criterion [133] for selecting features is to minimize the function:

$$\boldsymbol{S}_z = \frac{\sum_{ij} (\boldsymbol{f}_{zi} - \boldsymbol{f}_{zj})^2 \boldsymbol{G}_{ij}}{\boldsymbol{Var}(\mathbf{f}_z)} \tag{4.16}$$

where Var(**f**$_z$) is the variance for *z*-th feature, and f_{zi}, f_{zj} are *z*-th feature value for node *i*,*j*. Based on some simple calculation, we could obtain:

$$\sum_{ij}(f_{zi}-f_{zj})^2G_{ij}=2\sum_{ij}f_{zi}^2G_{ij}-2\sum_{ij}f_{zi}G_{ij}f_{zj}=2\mathbf{f}_z^TQ\mathbf{f}_z-2\mathbf{f}_z^TG\mathbf{f}_z=2\mathbf{f}_z^TL\mathbf{f}_z \quad (4.17)$$

By the spectral graph theory, the is calculated as:

$$Var(\mathbf{f}_z)=\sum_i\tilde{\mathbf{f}}_{zi}{}^2Q_{ii}=\tilde{\mathbf{f}}_z{}^TQ\tilde{\mathbf{f}}_z \quad (4.18)$$

Also, it is easy to show

$$\tilde{\mathbf{f}}_z{}^TL\tilde{\mathbf{f}}_z=\mathbf{f}_z^TL\mathbf{f}_z \quad (4.19)$$

Based on Eqs. (4.16)(4.17)(4.18)(4.19), the selection criteria Eq. (4.14) is evaluated. When all the features are assigned the score, the feature selection is carried out based on the score ranking.

4.2.3. Experimental evaluation

Experiments were conducted on ten data sets from UCI Repository to demonstrate the effectiveness of our feature ranking and selection method. There are six binary data sets and four multiple categorical data sets. In the experiment, each data set is randomly separated into two equal parts. One part is used in training and the rest part is used in testing. We used the training data to build the model for feature selection. Five filter feature selection models are utilized for comparison, including our proposed unsupervised filter model via ℓ_1 graph, Pearson correlation (supervised and unsupervised fashion), Fisher score (supervised filter model) and Laplacian score (unsupervised filter model). We use FL1, PCS, PCU, Fisher and Lap as abbreviations to denote these 5 methods in the experiments.

A simple case study for data “wine” is shown in Figure 4.8. Totally, there are 13 features for data wine, such as “Alcohol", “Magnesium" and “Proline". We use five filter methods based on training data (with size 89) and apply on the testing data. Each method chooses two features for 2-D visualization on testing data. Two features are selected by 5 different methods and plotted in each subfigure. It can be observed that the feature “Flavanoids" and feature “Color intensity" selected by FL1 method are crucial for discrimination.

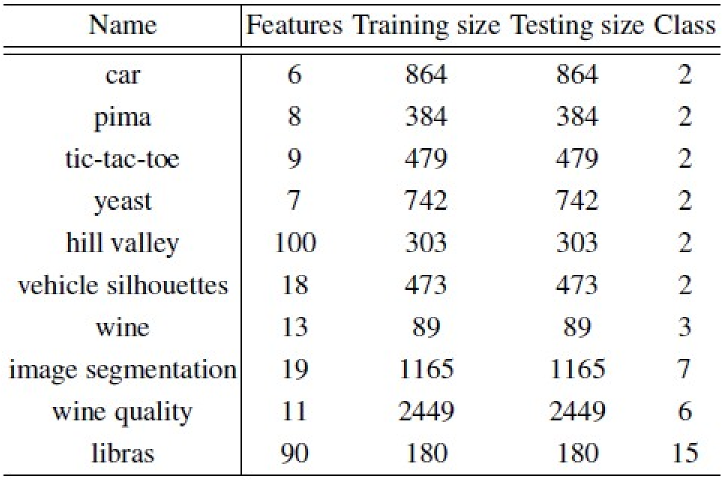
When selected features are more than two, we used the feature based classification to compare the feature selection methods. The experiment is conducted five times, and means outputs are obtained. The target selected features size is from one to around 80% of whole feature size to give comprehensive comparison. In order to show the classification performances, we use three classic classifiers: k nearest neighbor (*k*=5 in the experiment), LibSVM and multi-layer feed-forward networks. For briefness, we only plot two data sets (one binary-category data set and one multi-category data set) results. We abbreviate the classifiers as LibSVM, 5-NN and NeuralNet in the figures.

Figure 4.9 shows the comparison results for data “hill valley”. When the selected features size is smaller than 40, FL1 results rank first among all the competitors with all three classifiers. And

FL1 outputs rank second when the selected features size is larger than 40. In the case of multi-category data “libras” in Figure 4.10 the performances of FL1 rank first in most cases. It is important to note that the PCS and Fisher are supervised feature selection methods. And the performance of FL1 is competitive to PCS and Fisher in the most feature sizes.

In order to give comprehensive comparison of different feature selection methods on multiple data sets, the mean accuracy in low dimension (from feature size one to around 40% of whole feature sizes) are calculated based on each data set and each classifier. Table 4.5 shows the detail mean outputs and the comparison results. The highest accuracy for each classifier is highlighted. It can be observed that FL1 can win 6, 6 and 5 times of 10 data sets with LibSVM, 5-NN and NeuralNet respectively.

Table 4.4. UCI datasets used in the experiments



Name	Features	Training size	Testing size	Class
car	6	864	864	2
pima	8	384	384	2
tic-tac-toe	9	479	479	2
yeast	7	742	742	2
hill valley	100	303	303	2
vehicle silhouettes	18	473	473	2
wine	13	89	89	3
image segmentation	19	1165	1165	7
wine quality	11	2449	2449	6
libras	90	180	180	15

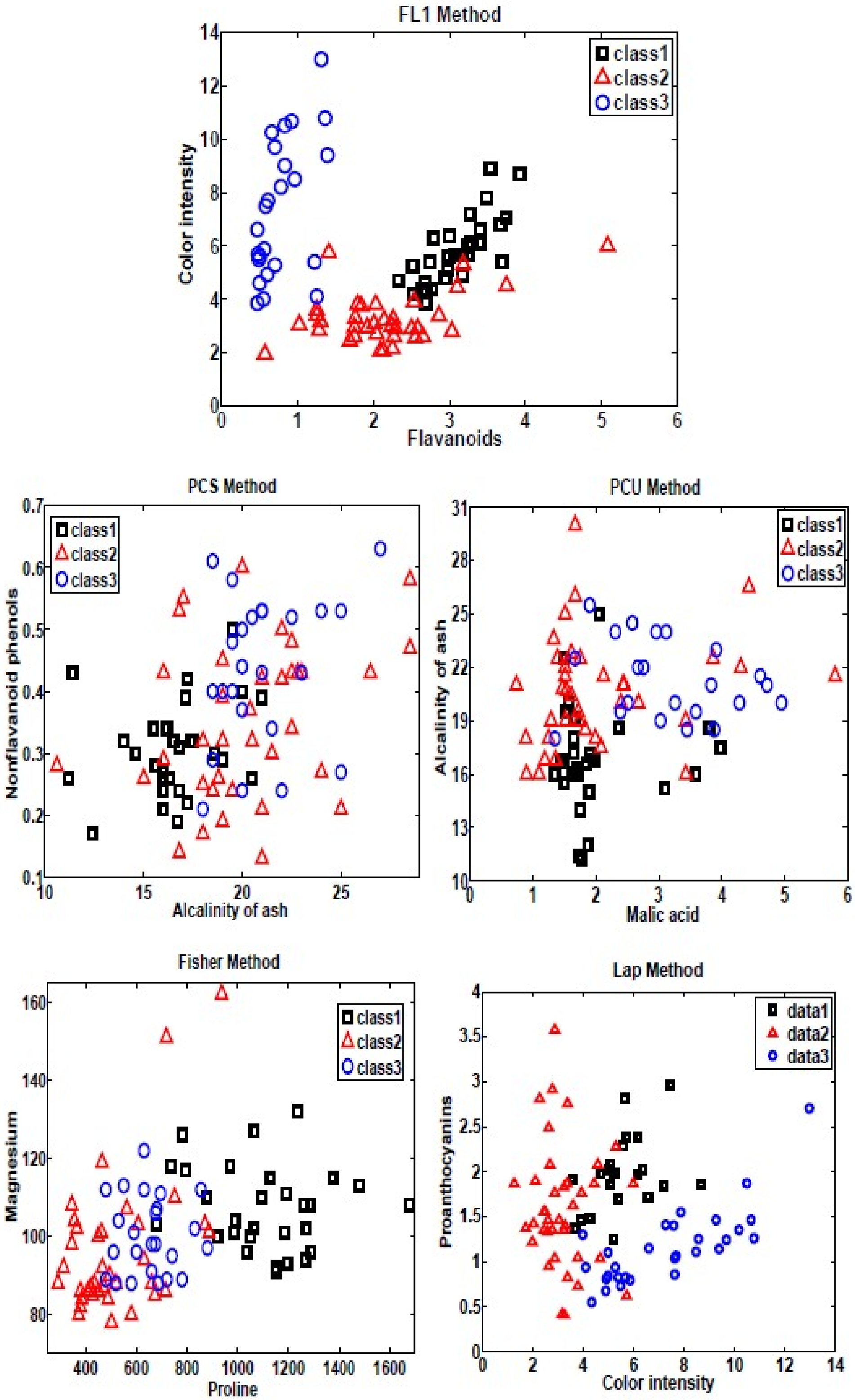


Figure 4.8. Data Wine plotted in 2-D with selected features. 5 methods have selected different two features.

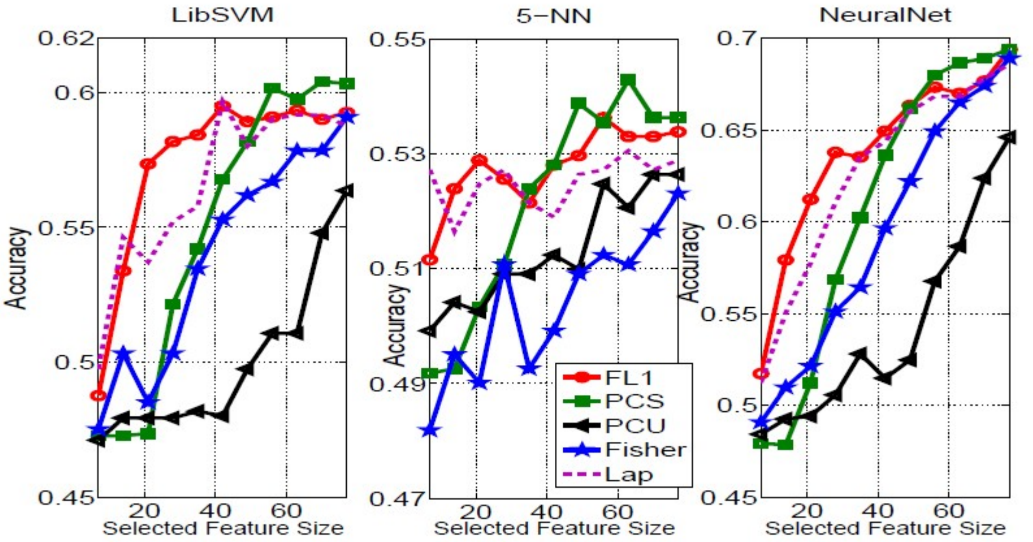


Figure 4.9. Comparison of feature based classification accuracies for data hill valley.

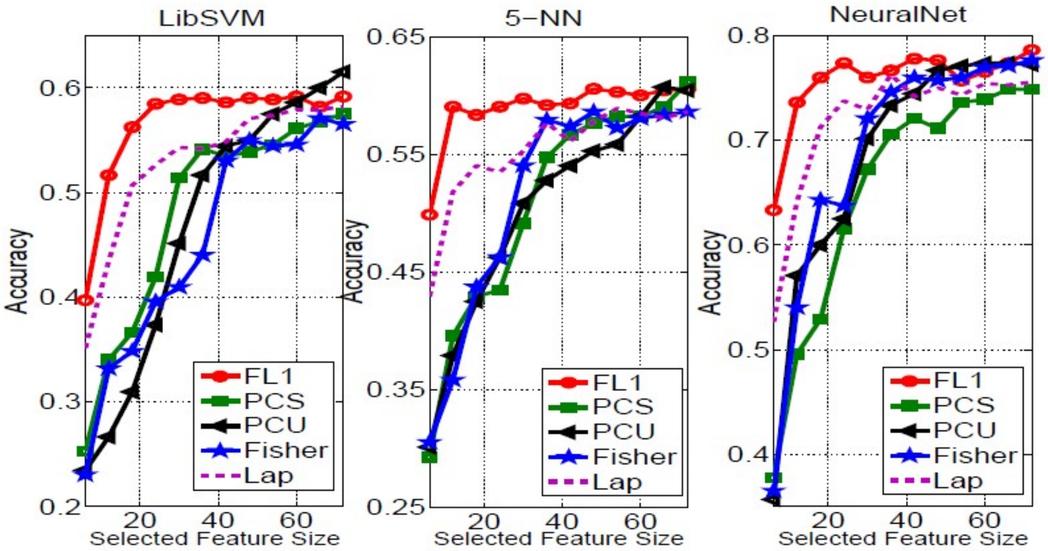


Figure 4.10. Comparison of feature based classification accuracies for data libras.

Table 4.5. Mean accuracy in low dimensions (in %)

Data set	LibSVM					5-NN					NeuralNet				
	FL1	PCS	PCU	Fisher	Lap	FL1	PCS	PCU	Fisher	Lap	FL1	PCS	PCU	Fisher	Lap
car	**73.7**	69.1	73.0	70.3	70.8	**70.7**	65.1	70.7	69.6	69.6	**74.9**	68.9	73.6	70.4	71.4
pima	**71.6**	64.9	65.3	67.8	67.3	**68.6**	61.4	64.0	64.6	65.8	**71.3**	64.6	66.8	67.5	67.3
tic-tac-toe	**63.4**	62.4	61.0	59.7	59.7	62.9	61.0	66.6	**68.9**	67.5	65.4	63.9	68.0	**71.6**	69.3
yeast	72.7	71.1	72.1	**72.8**	71.0	**70.7**	66.7	69.6	67.4	67.8	73.4	71.2	**73.8**	73.2	71.8
hill valley	**55.9**	50.8	47.9	50.9	54.8	**52.3**	50.8	50.6	49.4	52.1	**60.5**	54.6	50.3	53.9	58.8
vehicle	73.6	74.8	74.5	74.4	**75.0**	77.2	78.2	77.8	**78.6**	77.6	78.8	78.7	78.4	78.3	**79.4**
wine	**82.6**	69.8	66.6	75.1	75.4	**83.6**	69.3	68.7	79.4	75.5	**86.7**	72.8	71.0	79.4	77.7
image seg	91.1	91.3	90.8	90.1	**91.5**	92.8	**94.8**	92.0	91.2	93.7	93.4	**95.4**	92.4	91.6	94.1
wine quality	46.7	**55.7**	51.3	48.9	45.7	48.8	**52.9**	49.3	49.7	47.4	51.7	**55.8**	51.3	51.2	48.9
libras	**54.0**	40.6	35.8	35.9	48.4	**57.5**	43.2	43.4	44.7	52.5	**74.4**	56.6	60.0	61.1	68.2
Mean Wins	**6**	1	0	1	2	**6**	2	0	2	0	**5**	2	1	1	1

5. Conclusions and Discussions

5.1. Conclusions

5.1.1. Semantic Information Framework

In this project, we first introduced a cognitive linguistic (CL) based semantic information representation framework (SIRF).

Similar to text representations, SIRF encompasses representations at three abstraction levels, including lexical level, syntactic level and semantic level. Five cognitive linguistic conceptual primitives, i.e. thing, action, place, path and cause, are used across different levels to regularize the structures of the representations. At the lexical level, words in the forms of visual features are extracted and naturally mapped to different CL primitives. At the syntactic level, these primitives are further merged into phrases, or high level primitives. At the semantic level, these phrases are collected to form concepts.

The feature level primitive extraction can be accomplished using many existing image processing tools and methods. In this project we also introduced several new tools to address some less well studied challenges.

At the syntactic level, we proposed a probabilistic context free grammar (PCFG) approach to model the construction of syntactic visual phrases. Our method includes a grammar parsing component and rule induction component. In this work we focused on a special use case, i.e. small human group action recognition. The proposed can successfully addresses the concurrent sub-event problem in otherwise a linear grammar system. The semantic merge operation ensures that the representation is close to the human language as well as with a minimum description length. The experimental results in small human group event recognition applications demonstrate the effectiveness of our proposed framework. Although we utilized some domain-specific knowledge in this application, the general representation can be applied to many different visual recognition systems.

At the semantic level, we proposed a Bayesian network (BN) approach to facilitate concept prediction and inference. With the help of prior knowledge, a BN can be constructed with appropriate CL primitives for each concept. The causal relationships are usually intuitive to establish, and their conditional probability distributions of these CL primitives can be estimated through data training. Given a constructed BN, or SIRF concept model (SCM), we can make inference on the concept based on the observations. A simple use case of “small human group actions” was presented to demonstrate the SCM construction and inference procedures.

5.1.2. Primitive Modeling

In the second phase of this project, we developed several CL primitive modeling methods to improve primitive extraction in some very challenging scenarios.

We proposed a dynamic structure preserving map (DSPM) as a spatio-temporal model to recognize individual human actions from video sequences. It is an extension to a special type of neural network called self organizing map (SOM). Through learning on low-level features, DSPM automatically extracts intrinsic spatio-temporal patterns from the video sequence. DSPM improves the adaptive learning rule in SOM with a Markov model on the dynamic behavior of best matching units, which helps to preserve spatio-temporal dynamic topological structure. Through the non-linear mapping, DSPM can reduce computational cost and data redundancy for action recognition. The ensemble learning based on EM is adopted to estimate the latent parameters. Our experimental results on several popular human action datasets showed that DSPM is a very effective and competitive method in human action recognition.

We further introduced a novel structural feature to describe small human group actions. The feature set is derived from social network analysis (SNA). The major advantage of this SNA based feature set is that it can capture group structure while ignoring many unnecessary specifics of individuals involved. Therefore this feature set can handle group actions with varying numbers of individuals, varying time durations, and individual occlusions. Based on this feature, we also propose a Conditional Gaussian Process Dynamic Model (GPDM) for dynamic modeling. We demonstrated competitive and robust results on the group human activity recognitions by constructing middle level features only on position cues, and by using models trained from one data set to test on an entirely different dataset.

Besides small group activities, we also extended our SNA based GPDM method for recognizing human object interactions. In this work, key human body parts and the object are considered as nodes in a social network graph. A special set of social network analysis based features is introduced to capture the distributions of motion patterns among all the nodes overtime. It provides a global view of the activity while preserving the individuality of each node. Because of these, our method can tolerate missing information of the low-level detections on human body parts and the small object. We have shown that this method can achieve good performance in very challenging human object interaction scenarios.

Although human object recognition has been extensive studied, most of the existing methods are based on sophistical appearance models. While working on our surveillance related video sequences, we realized that human objects are usually too small for recognition. Therefore we introduced a new human object recognition methods based on a less explored human biometric, human silhouette sequence. In this work, we introduced a binary silhouette subtraction and pHOG features based human gait recognition method. The Pyramid of HOG feature on the binary silhouette images effectively captures the shape dynamics during the motion of the human object. Our experimental results verified that the proposed method can achieve a competitive recognition rate in comparison with other existing human gait recognition methods.

5.1.3. Sensor Utility Metrics

In the third phase of the project, we developed two new feature selection method based under that sparse coding framework.

First we introduced an unsupervised filter model Feature Selection via Sparse Imputation (FSSI). In particular, the imputation quality of individual feature is utilized as evaluation metrics in feature selection. The proposed method is applied to UCI data sets (binary-category and multiple-category). The classification results are obtained with classic classifiers (support vector machine, k nearest neighbors and multi-layer feed forward networks). The proposed feature selection method was compared with other popular methods, including Fisher score method (a supervised filter model) and Laplacian score method (a unsupervised filter model). The comparison results on UCI data sets demonstrate the capability and efficiency of our method.

We also extended the FSSI idea and developed another supervised filter model feature selection method based on the ℓ_1 -graph representation of sparse coding. Our approach aims to use ℓ_1 -graph to evaluate the local property for individual feature. A similarity matrix based on sparse subspace clustering was defined to construct ℓ_1 -graphs. The feature's local preserving ability is evaluated through spectral graph theory. Experimental comparisons with related filter methods have demonstrated that our method is effective in terms of visualization and classification.

5.2. Future Directions

During the course of this project, we have acquired valuable insight on semantic sensor information representation. Some of the possible future research directions are outlined as follows.

1. Similar to the level of text representations, we believe that a comprehensive semantic sensor information representation requires all three levels of representations, i.e. signal level, syntactic level, and semantic level. The main objective of signal level representation is acquisition, the main objective of syntactic level representation is compression/summarization/abstraction, and the main objective of semantic level representation is reasoning. It is not advisable to seek a single mathematical model for all three levels. Decades of research on signal processing and computer vision has made significant advances on signal level representations. Recently, syntactic level sensor information representations have attracted substantial amount of attentions, and many effective methods have be proposed. At semantic level, although there have been many tools developed for expert systems, decision support systems and database systems etc., they are mainly text based systems, and they are not well suited to sensor syntactic analysis outputs. Our proposal of a Bayesian network with Probabilistic Context Free Grammar provides one potential solution to this challenge. However we understand that we have not entirely solved this problem, which is obvious when we look at an SCM XML file from a human user (page 11) and an output SCM XML file from our parsing system (page 12). We will continue to explore a seamless integration of a parsing model and a reasoning model.

2. We realize that a major challenge in high dimensional sensor information representation is the difficulty in structure preserving. In a general sense, “structure” refers to relative spatial and temporal relationships among various information elements. Some of the most effective detection and classification methods have been based on “bag of words” or “bag of features” approach. They are highly efficient in computation, but deficient in representation. Their performance heavily relies on appropriate training. On the other hand, structure preserving methods usually suffer low robustness. In this project we intentionally focused on improving efficiency and robustness of structure preserving methods. At signal level, we explored SNA based features, which are able to preserve group structures without the detail specifications of individuals. We also applied SOM based feature clustering method that is able to discover certain topological relationships in the feature space. At syntactic level, we studied PCFG as a generative model that can preserve well defined structures, such as shapes and trajectories. We are currently exploring a systematic extension of PCFG to a multi-dimensional signal space.

3. Our original proposal was motivated by advances in cognitive linguistics. Due to the scope of this project, we could only explore a very small portion of cognitive linguistics principles. A major hypothesis of cognitive linguistics “experiential embodiment”, which suggests that language is not a set of symbols with rules, but is a pointer to **shared prior experience**, i.e. language is a context-sensitive index set. Computational semanticists claim that meaning consists of relationships among symbols. Cognitive linguists point out that sensorimotor experience with an unknown object is much richer internally than the words that describe it. Cognitive linguists argue that the primary primate reasoning mechanism is metaphor, the binding of new experiences and words to existing personal sensorimotor experiences and related words. We believe that cognitive linguistics can have great potentials in sensor information processing. For example, cognitive linguistics is particular suited for active sensing, when sensing system acquire knowledge through sensorimotor experience. Also cognitive linguistics can establish an effective interface between human query in natural language and sensing system outputs through experience of interactions. Furthermore, computational models of metaphoric reasoning can provide natural solutions for cross-modality data fusion as well as soft data and hard data fusion.

5.3. Summary of achievements on SOW tasks:

Task 1: formulating mathematical and theoretical foundations of Semantic Information Representation Framework (SIRF)

- We have introduced a cognitive linguistics based semantic information representation framework (SIRF). We defined its components, layered architecture, and data structure.
- We have demonstrated that, at a low abstraction level in SIRF, syntactic constructs can be learned from observations probabilistic context-free grammar (PCFG).
- We have demonstrated that, at a high abstraction layer in SIRF, semantic reasoning (abstraction and inference) can be achieved using Bayesian networks

Task 2: development of model distributions for analysis and information fusion

- We have developed a novel Dynamic Structure Preserving Map (DSPM) method for individual human action primitive modeling
- We have developed a small human group activity modeling method based on Gaussian Process Dynamic Model and social network analysis (SN-GPDM)
- We have extended SN-GPDM method to recognize human object interactions
- We have developed a human gait recognition method based on pyramid histogram of oriented gradients (pHOG) on optical flow features.

Task 3: development of quantitative metrics of semantic utility

- We have developed a unsupervised feature selection via sparse imputation (FSSI) method to determine the importance of individual features and sensor modalities
- We have generalized individual feature contributions in sparse representation into L-1 graphs, which can be used in feature selection and utility ranking.

Publications:

Journals:

- Yuan Cao, Haibo He, and Hong Man, " SOMKE: Kernel Density Estimation over Data Streams Based on Self-Organizing Map", *IEEE Trans. Neural Networks*, May 2012.
- Jin Xu, Haibo He, and Hong Man, "DCPE Co-Training for Classification", *Neurocomputing* (Elsevier), vol. 86, pp. 75-85. 2012.
- Xiaopeng Huang, Ravi Netravali, Hong Man and Victor Lawrence, "Multi-Sensor Fusion of Infrared and Electro-Optic Signals for High Resolution Night Images", *Sensors*, 2012, 12(8), 10326-10338; doi:10.3390/s120810326
- Qiao Cai, Haibo He, and Hong Man, "Spatial Outlier Detection Based on Iterative Self-Organizing Learning Model ", *Neurocomputing* (Elsevier), Vol. 117 Pg. 161-172, 2013.
- Jin Xu, Guand Yang, Yafeng Yin, Hong Man and Haibo He, "Sparse Representation Based Classification with Structure Preserving Dimension Reduction", *Cognitive Computation* (Springer), Feb. 2013 (minor revision)
- Jin Xu, Haibo He, and Hong Man, "Semi-Supervised Feature Selection Based on Relevance and Redundancy Criteria", *IEEE Trans. Neural Networks and Learning Systems*, Mar. 2013 (submitted)
- Qiao Cai, Haibo He, and Hong Man, "Imbalanced Evolving Self-Organizing Learning", *Neurocomputing* (Elsevier), Mar. 2013 (submitted)

Conferences:

- Jin Xu, Yafeng Yin, Hong Man and Haibo He, “Feature Selection Based on Sparse Imputation”, *International Joint Conference on Neural Networks* (IJCNN), Brisbane, Australia, June 2012.
- Yafeng Yin, Guang Yang, Jin Xu, Hong Man, "Small Group Human Activity Recognition", *IEEE International Conference on Image Processing* (ICIP), Orlando, FL, September 2012.
- Guang Yang, Yafeng Yin, Jeanrok Park and Hong Man, “Human Gait Recognition by Pyramid of HOG feature on Silhouette Images”, *SPIE Defense, Security and Sensing 2013*, Baltimore MD, April 2013.
- Xiaopeng Huang, Ravi Netravali, Hong Man, Victor B. Lawrence, “Multisensor fusion of electro-optic and infrared signals for high-resolution visible images”, *SPIE Defense, Security and Sensing 2013*, Baltimore MD, April 2013.
- Qiao Cai, and Hong Man, "DSPM: Dynamic Structure Preserving Map for Action Recognition ", *IEEE International Conference on Multimedia and Expo* (ICME), San Jose CA, July 2013. (acceptance rate: 12%, best paper candidate)
- Jin Xu, Guang Yang, Hong Man, and Haibo He, "L1-Graph Based on Sparse Coding for Feature selection", *International Symposium on Neural Networks* (ISNN 2013), July 2013.

- Guang Yang, Yafeng Yin, and Hong Man, "Small Human Group Detection and Event Representation Based on Cognitive Semantics ", *IEEE International Conference on Semantic Computing* (ICSC), Irvine CA, Sep. 2013 (acceptance rate: 30%)
- Qiao Cai, Yafeng Yin, and Hong Man, "Learning Spatio-temporal Dependencies for Action Recognition", *IEEE International Conference on Image Processing* (ICIP), Melbourne Australia, Sep. 2013
- Guang Yang, Yafeng Yin, and Hong Man, "Recognizing Interactions Between Human and Object Based on Social Network Analysis", *IEEE Applied Imagery Pattern Recognition Workshops* (AIPR), Washington DC, Oct. 2013

Book Chapter:

- Yafeng Yin, and Hong Man, "Behavior Modeling of Human Objects in Multimedia Content", invited chapter in Multimedia Security and Steganography, Ed. Frank Shih (CRC Press), Oct 2011.

References

[1] W. Croft and D. A. Cruse, *Cognitive Linguistics*, Cambridge University Press, 2004.

[2] George Lakoff, *Women, Fire, and Dangerous Things*, Prentice Hall, 1987.

[3] R. Jackendoff, *Semantic Structures*, Mit Press, 1993.

[4] M. Johnson and T. Rohrer, "We are live creatures: Embodiment, American Pragmatism," in *Body, Language, Mind: Embodiment*, T. Ziemke et al, Editors, (Berlin: Mouton de Gruyter), 2007.

[5] Peter Gärdenfors, "Representing actions and functional properties in conceptual spaces," in *Body, Language, Mind: Embodiment*, T. Ziemke et al, Editors, (Berlin: Mouton de Gruyter) 2007

[6] J. Mitola and H. Man, "Semantics in cognitive radio," in *Third IEEE International Conference on Semantic Computing* (ICSC), Berkeley, CA, Sept. 2009.

[7] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, vol. 284, no. 5, pp. 34–41, 2001.

[8] A. Caramazza, "How Many Levels of Processing Are There in Lexical Access?", Cognitive Neuropsychology, 14(1), 177-208, 1997.

[9] C. Manning and C. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[10] K. Lari and S. J. Young, "The Estimation of Stochastic Context Free Grammars Using the Inside-Outside Algorithm", *Computer Speech and Language*, vol. 4, pp35-36, 1990.

[11] M. I. Jordan, *Learning in Graphical Models*, MIT Press, 1999.

[12] N. Chomsky, "Three Models for the Description of Language", IRE Transactions on Information Theory (2), 113-124, 1956.

[13] M. S. Ryoo and J. K. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Volume 2, Washington, DC, USA, 2006, pp. 1709–1718.

[14] Vlad I. M. and Larry.S. D, "Multi-agent event recognition in structured scenarios," in Computer Vision and Pattern Recognition, The 24th IEEE Conference on, Colorado Springs, CO, USA, Jun. 2011, pp. 3289–3296.

[15] Z Si, M Pei, B. Yao, and S Zhu, "Unsupervised learning of event and-or grammar and semantics from video," in Computer Vision, 2011 IEEE International Conference on, Barcelona, Spain, Nov. 2011, pp. 41–48.

[16] S. Joo and R. Chellappa, "Attribute grammar-based event recognition and anomaly detection," in Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, USA, Jun. 2006, pp. 107–114.

[17] Z. Zhang, T. Tan, and K. Huang, "An extended grammar system for learning and recognizing complex visual events," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 2, pp. 240–255, 2011.

[18] B. Jin, W. Hu, and H. Wang, “Human interaction recognition based on transformation of spatial semantics,” IEEE Signal Processing Letters, vol. 19, no. 3, pp. 139–142, Mar. 2012.

[19] J. D. Cruz, C. Bothorel, and F. Poulet, “Entropy based community detection in augmented social networks,” in Computational Aspects of Social Networks (CASoN), 2011 International Conference on, oct. 2011, pp. 163–168.

[20] Y. Yin, G. Yang, J. Xu, and H. Man, “Small human group activity recognition,” in Image Processing, the 2012 IEEE International Conference on, Orlendo, FL, USA, oct. 2012, pp. 2709–2712.

[21] W. Ge, Collins R. T., and Ruback R. B., “Vision-based analysis of small groups in pedestrian crowds,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 5, pp. 1003–1016, 2012.

[22] Jorma Rissanen, “A universal prior for integers and estimation by minimum description length,” The Annals of Statistics, vol. 11, no. 2, pp. 416–431, 1983.

[23] Grnwald Peter, “A minimum description length approach to grammar inference,” in Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing. 1995, vol. 1040 of Lecture Notes in Computer Science, pp. 203–216, Springer.

[24] A. Stolcke, “An efficient probabilistic context-free parsing algorithm that computes prefix probabilities,” Computational Linguistics, vol. 21, no. 2, pp. 165–201, Jun. 1995.

[25] S. Blunsden and R. B. Fisher, “The behave video dataset: ground truthed video for multi-person,” Annals of the BMVA, vol. 4, pp. 1–12, 2010.

[26] W. Choi, K. Shahid, and S. Savarese, “What are they doing? : Collective activity classification using spatiotemporal relationship among people,” in IEEE 12th International Conference on Computer Vision Workshops, Kyoto, Japan, oct. 2009, pp. 1282–1289.

[27] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” IEEE Trans. Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1473–1488, 2008.

[28] H. Lee, R. Grosse, R. Ranganath, and A. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” ICML, 2009.

[29] T. Duong, H. Bui, D. Phung, and S. Venkatesh, “Activity recognition and abnormality detection with the switch- ing hidden semi-markov model,” CVPR, pp. 838–845, 2005.

[30] T. Kohonen, *Self Organizing Maps*, Springer, 2001.

[31] A. Yilmaz and M. Shah, “Actions sketch: a novel action representation,” CVPR, 2005.

[32] I. Laptev, “On space-time interest points,” Intl. Journal of Computer Vision, vol. 64, pp. 107–123, 2005.

[33] G. Willems, T. Tuytelaars, and L. Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” ECCV, 2008.

[34] Y. Lui, J. Beveridge, and M. Kirby, “Action classification on product manifolds,” CVPR, 2010.

[35] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li, “Hierarchical spatio-temporal context modeling for action recognition,” CVPR, 2009.

[36] A. Gilbert, J. Illingworth, and R. Bowden, “Fast realistic multi-action recognition using mined dense spatiotemporal features,” ICCV, 2009.

[37] Y. Ke, R. Sukthankar, and M. Hebert, “Spatiotemporal shape and flow correlation for action recognition,” CVPR, 2007.

[38] Y. Zhu, X. Zhao, Y. Fu, and Y. Liu, “Sparse coding on local spatial-temporal volumes for human action recognition,” ACCV, pp. 660–671, 2010.

[39] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” ICPR, 2004.

[40] X. Wu, D. Xu, L. Duan, and J. Luo, “Action recognition using context and appearance distribution features,” CVPR, 2011.

[41] A. Fathi and G. Mori, “Action recognition by learning mid-level motion features,” CVPR, 2008.

[42] A. Kovashka and K. Grauman, “Learning a hierarchy of discriminative space-time neighborhood features for human action recognition,” CVPR, 2010.

[43] M. Varsta, J. Heikkonen, J. Lampinen, and J. Millan, “Temporal kohonen map and recurrent self-organizing map: analytical and experimental comparison,” Neural Processing Letters, vol. 13, pp. 237–251, 2001.

[44] M. Szummer and T. Jaakkola, “Partially labeled classification with markov random walks,” NIPS, 2001.

[45] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” CVPR, 2005.

[46] M. Rodriguez, J. Ahmed, and M. Shah, “Action mach: a spatio-temporal maximum average correlation height filter for action recognition,” CVPR, 2008.

[47] D. Fleet and Y. Weiss, “Optical flow estimation,” Handbook of Mathematical Models in Computer Vision, Springer, pp. 239–258, 2005.

[48] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, “A biologically inspired system for action recognition,” ICCV, 2007.

[49] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos “in the wild”,” CVPR, 2009.

[50] J. C. Niebles, H.Wang, and F. Li, “Unsupervised learning of human action categories using spatial-temporal words,” in BMVC 2006.

[51] S.-F. Wong and C. Roberto, “Extracting spatiotemporal interest points using global information,” in ICCV 2007.

[52] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in CVPR 2008.

[53] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009.

[54] X. Wang, X. Ma, and W. Grimson, “Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 3, pp. 539 –555, 2009.

[55] B. Ni, S. Yan, and A. Kassim, “Recognizing human group activities with localized causalities,” in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009.

[56] M.-C. Chang, N. Krahnstoever, S. Lim, and T. Yu, “Group level activity recognition in crowded environments across multiple cameras,” in Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2010.

[57] W. Ge, R. T. Collins, and B. Ruback, “Automatically detecting the small group structure of a crowd,” in Workshop on Applications of Computer Vision, WACV 2009.

[58] R. Guimera, B. Uzzi, J. Spiro, and L. A. N. Amaral, “Team assembly mechanisms determine collaboration network structure and team performance,” in Science, vol. 308, no. 5722, 2005, pp. 697 – 702.

[59] L. Raskin, M. Rudzsky, and E. Rivlin, “Tracking and classifying of human motions with gaussian process annealed particle filter,” in ACCV07.

[60] J. M. Wang, D. J. Fleet, and A. Hertzmann, “Gaussian process dynamic models for human motion,” IEEE Transactions on Pattern Analysis And Machine Intelligence, vol. 30, pp. 283–298, 2008.

[61] J. Wang, Y. Yin, and H. Man, “Multiple Human Tracking Using Particle Filter with Gaussian Process Dynamical Model”, EURASIP Journal on Image and Video Processing, Special Issue on 3D Image and Video Processing, Volume 2008 (2008).

[62] R. T. Collins, Y. Liu, and M. Leordeanu, “Online selection of discriminative tracking features,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1631 –1643, 2005.

[63] K. Smith, D. Gatica-perez, and J.-m. Odobez, “Using particles to track varying numbers of interacting people,” in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2005.

[64] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” 2007.

[65] D. J. Watts and D. H. Strogatz, “Collective dynamics of ’small-world’ networks,” Nature, vol. 393, no. 6684, pp. 440–442, June 1998.

[66] S. Wasserman and K. Faust, Social Networks Analysis: Methods and Applications. Cambridge: Cambridge University Press., 1994.

[67] V. Krebs, “The social life of routers,” in Internet Protocol Journal, Dec. 2000, pp. 14–25.

[68] N. Lawrence, “Probabilistic non-linear principal component analysis with Gaussian process latent variable models,” Journal of Machine Learning Research, vol. 6, pp. 1783–1816, 2005.

[69] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, pp. 2323–2326, 2000.

[70] J. B. Tenenbaum, “Mapping a manifold of perceptual observations,” 1998, pp. 682–688, advances in Neural Information Processing Systems, 10.

[71] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-fei, “Unsupervised learning of human action categories using spatial-temporal words,” in Proceedings of British Machine Vision Conference, BMVC 2006.

[72] Shu-Fai Wong and Roberto Cipolla, “Extracting spatiotemporal interest points using global information,” in IEEE 11th International Conference on Computer Vision, ICCV 2007.

[73] Y. Yin, G. Yang, J. Xu, and H. Man, “Small group human activity recognition,” in 2012 19th IEEE International Conference on Image Processing, (ICIP), Sept. 2012.

[74] C. Desai, D. Ramanan, and C. Fowlkes, “Discriminative models for static human-object interactions,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2010, pp. 9 –16.

[75] Bangpeng Yao and Li Fei-Fei, “Recognizing humanobject interactions in still images by modeling the mutual context of objects and human poses,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pp. 1691 –1703, sept. 2012.

[76] A. Prest, C. Schmid, and V. Ferrari, “Weakly supervised learning of interactions between humans and objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, pp. 601 –614, march 2012.

[77] A. Gupta, A. Kembhavi, and L.S. Davis, “Observing human-object interactions: Using spatial and functional compatibility for recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 10, pp. 1775 –1789, oct. 2009.

[78] A. Prest, V. Ferrari, and C. Schmid, “Explicit modeling of human-object interactions in realistic videos,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PP, no. 99, pp. 1, 2012.

[79] Zhangzhang Si, Mingtao Pei, B. Yao, and Song-Chun Zhu, “Unsupervised learning of event and-or grammar and semantics from video,” in 2011 IEEE International Conference on Computer Vision (ICCV), nov. 2011, pp. 41 –48.

[80] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang, “Real-time compressive tracking,” in ECCV, 2012, pp. 864–877.

[81] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: a large video database for human motion recognition,” in Proceedings of the International Conference on Computer Vision (ICCV), 2011.

[82] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in Computer Vision and Pattern Recognition (CVPR), june 2011, pp. 1385 –1392.

[83] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627 –1645, Sept. 2010.

[84] R. Fergus and P. Perona, “Caltech object category datasets,” http://www.vision.caltech.edu/ html-files/archive.html.

[85] Duncan J. Watts and Duncan H. Strogatz, “Collective dynamics of ’small-world’ networks,” Nature, vol. 393, no. 6684, pp. 440–442, June 1998.

[86] Han, J. and Bhanu, B., “Individual recognition using gait energy image,” in [IEEE Trans. Pattern Anal. Mach. Intell.], 28, 316–322 (Feb. 2006).

[87] Nizami, I. F., Hong, S., Lee, H., Lee, B., and Kim, E., “Automatic gait recognition based on probabilistic approach,” in International Journal of Imaging Systems and Technology.

[88] Bashir, K., Xiang, T., and Gong, S., “Gait recognition without subject cooperation,” in Pattern Recognition Letters, 31, 2052–2060 (October 2010).

[89] Sarkar, S., Phillips, P., Liu, Z., Vega, I., Grother, P., and Bowyer, K., “The humanid gait challenge problem: data sets, performance, and analysis,” in IEEE Trans. Pattern Anal. Mach. Intell., 27, 162 –177 (Feb. 2005).

[90] Dockstader, S. L., Bergkessel, K. A., and Tekalp, A. M., “Feature extraction for the analysis of gait and human motion,” in [Proceedings of ICPR’02 Volume 1 - Volume 1], ICPR ’02, 10005– (2002).

[91] Cheng, M.-h., Ho, M.-f., and Huang, C.-l., “Gait analysis for human identification through manifold learning and hmm,” in Pattern Recognition, 41, 2541–2553 (2008).

[92] Dalal, N. and Triggs, B., “Histograms of oriented gradients for human detection,” in CVPR’05, 2, 886–893 (June 2005).

[93] Horprasert, T., Harwood, D., and Davis, L. S., “A robust background subtraction and shadow detection,” in In Proceedings of ACCV, (2000).

[94] Zheng, S., Zhang, J., Huang, K., He, R., and Tan, T., “Robust view transformation model for gait recognition,” in IEEE ICIP 2011, 2073 –2076 (2011).

[95] P. Mitra, C.A. Murthy and S.K. Pal, “Unsupervised feature selection using feature similarity,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 301-312, 2002.

[96] J. Mutch and D.G. Lowe, “Multiclass object recognition with sparse, localized features,” Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 11-18, 2006.

[97] H. Peng, F. Long and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226-1238, 2005.

[98] J.B. Yang and C.J. Ong, “Feature Selection using Probabilistic Prediction of Support Vector Regression,” IEEE Transactions on Neural Networks, vol. 22, no. 6, pp. 954-962, 2011.

[99] M. Hall, “Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning,” Proc. 17th Int’l Conf. Machine Learning, pp. 359-366, 2000.

[100] L. Wang, “Feature Selection with Kernel Class Separability,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 9, pp. 1534-1546, 2008.

[101] T. Pavlenko, “On feature selection, curse-of-dimensionality and error probability in discriminant analysis,” Journal of Statistical Planning and Inference, vol. 115, no. 2. pp. 565-584, 2003.

[102] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” Journal of Machine Learning Research 3, pp. 1157-1182, 2003.

[103] X. He, D. Cai and P. Niyogi, “Laplacian score for feature selection,” Proc. Adcenaces in the Neural Information Processing Systems 18, Vancouver, Canada, 2005.

[104] F. Wang and P. Li, “Compressed Nonnegative Sparse Coding,” IEEE 10th International Conference on Data Mining (ICDM), pp. 1103-1108, 2010.

[105] Y. Li, S. Amari, A. Cichocki, D.W.C. Ho, and S. Xie, “Underdetermined blind source separation based on sparse representation,” IEEE Transactions on Signal Processing, vol.54, no.2, pp. 423-437, 2006.

[106] M. Elad, and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Transactions on Image Processing, 15(12), pp. 3736-3745, 2006.

[107] J. Wright, A.Y. Yang, A. Ganesh, S.S Sastry and Y. Ma, “Robust face recognition via sparse representation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(2), pp. 210-227, 2009.

[108] J. Gemmeke and B. Cranen, “Using sparse representations for missing data imputation in noise robust speech recognition,” Proc. of EUSIPCO 2008.

[109] A. Frank and A. Asuncion, “UCI Machine Learning Repository,” [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science. 2010

[110] R.O. Duda P.E. Hart and D.G. Stork, “Pattern Classification,” Wiley- Interscience Publication, 2001.

[111] K. Tsuda, M. Kawanabe and K.R. Muller, “Clustering with the Fisher Score”, Advances in Neural Information Processing Systems 15, MIT Press, 2003.

[112] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection,” International Conference on Uncertainty in Artificial Intelligence, 2011.

[113] M. Belkin and P. Niyogi, “Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering,” Advances in Neural Information Processing Systems, vol. 14, 2001.

[114] X. He and P. Niyogi, “Locality Preserving Projections,” Advances in Neural Information Processing Systems, vol. 16, 2003.

[115] E. Candes and T. Tao, “Near optimal signal recovery from random projections and universal encoding strategies,” IEEE Trans. Inform. Theory, vol. 52, pp. 5406-5225, 2006.

[116] S. Kim, K. Koh, M. Lustig, S.Boyd and D. Gorinevsky, “An interiorpoint method for largescale l1-regularized least squares”, IEEE Journal of Selected Topics in Signal Processing, 1(4), pp. 606-617, 2007.

[117] S. Mallat and Z. Zhang, “Matching pursuit with time-frequency dictionaries”, IEEE Trans. Signal Process, vol. 41, pp.3397-3415, 1993.

[118] S.S. Chen, D.L. Donoho, and M.A. Saunders,“Atomic decomposition by basis pursuit”, SIAM Journal on Scientific Computing, 20(1), pp. 33-61, 1998.

[119] J. Mairal, F. Bach, J. Ponce and G. Sapiro, “Online Dictionary Learning for Sparse Coding”, International Conference on Machine Learning, 2009.

[120] J. Xu and H. Man, “Dictionary Learning Based on Laplacian Score in Sparse Coding,” Lecture Notes in Computer Science, vol. 6871, pp. 253-264, 2011.

[121] P.D. Allison, “Multiple Imputation for Missing Data: A Cautionary Tale,” Sociological Methods and Research, vol. 28, pp. 301-309. 2000.

[122] Y. Freund and R.E.Schapire, “A decision-theoretic generalization of online learning and an application to boosting,” Journal of computer and system sciences 55, pp. 119-139, 1997.

[123] P. Silapachote, D.R. Karuppiah and A. Hanson, “Feature selection using adaboost for face expression recognition,” The Fourth IASTED International Conference on Visualization, Imaging, and Image Processing, Marbella, Spain, pp. 84-89, 2004.

[124] L. Shen and L. Bai, “AdaBoost Gabor feature selection for classification,” Image and Vision Computing NewZealand (IVCNZ), Akaroa, New Zealand, pp. 77-83, 2004.

[125] Z. Zhao and H. Liu, “Semi-supervised feature selection via spectral analysis,” Proc. 7th SIAM International Conference on Data Mining, 2007.

[126] C.C. Chang and C.J. Lin, “LIBSVM: a library for support vector machines,” Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.

[127] I.H. Witten and E. Frank, Data mining: Practical Machine Learning Tools and Techniques with Java Implemeentations, San Francisco: Morgan Kaufmann, 2000.

[128] Y. Lee, Y. Lin and G. Wahba, “Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data,” Journal of the American Statistical Association, vol. 99, pp. 67-81, 2004.

[129] J. Ren, Z. Qiu, W. Fan, H. Cheng and P.S. Yu, “Forward semisupervised feature selection,” Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining (PAKDD), Osaka, Japan, 2008.

[130] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” Journal of Machine Learning Research 3, pp. 1157-1182, 2003.

[131] E. Elhamifar and R. Vidal, “Sparse Subspace Clustering”, IEEE International Conference on Computer Vision and Pattern Recognition, 2009.

[132] Fan R. K. Chung, “Spectral Graph Theory”, Regional Conference Series in Mathematics, no. 92, 1997.

[133] X. He, D. Cai and P. Niyogi, “Laplacian score for feature selection,” Proc. Adcenaces in the Neural Information Processing Systems 18, Vancouver, Canada, 2005